Encryption device, encryption method, decryption device, and decryption method

, Takanori; , Harunaga; , Andrey

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

Citation (APA):
T., H., & A. (2019). Encryption device, encryption method, decryption device, and decryption method. (Patent No. *WO2019031025*).

---

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2019年2月14日(14.02.2019)

WIPO | PCT

(10) 国際公開番号
WO 2019/031025 A1

(51) 国際特許分類:
G09C 1/00 (2006.01)

(21) 国際出願番号： PCT/JP2018/020341

(22) 国際出願日： 2018年5月28日(28.05.2018)

(25) 国際出願の言語： 日本語

(26) 国際公開の言語： 日本語

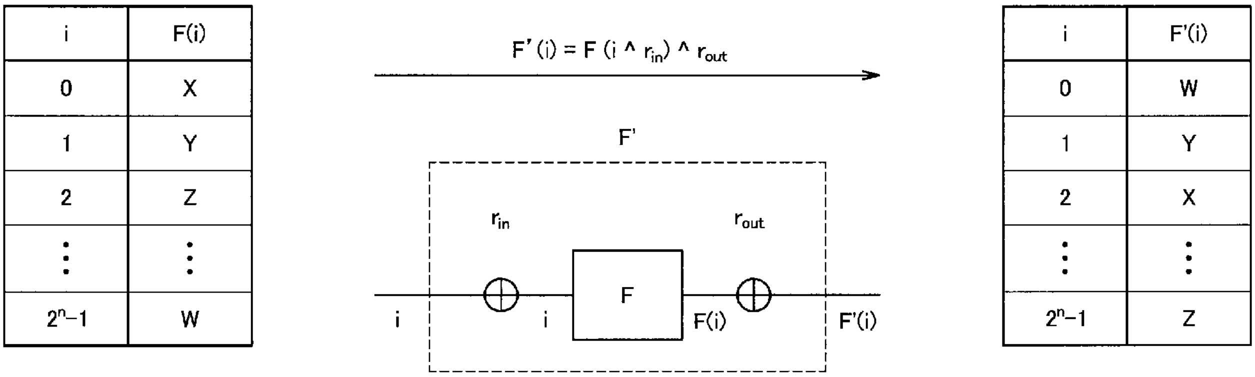
(30) 優先権データ：
特願 2017-156144 2017年8月10日(10.08.2017) JP

(71) 出願人: ソニー株式会社 (**SONY CORPORATION**) [JP/JP]; 〒1080075 東京都港区港南１丁目７番１号 Tokyo (JP). テクニカル ユニバーシティ オブ デンマーク(**TECHNICAL UNIVERSITY OF DENMARK**) [DK/DK]; 2800 コーゲーエス リュンビュー アンケル エンエルンズバイ １ ビルディング１０１エー Lyngby (DK).

(72) 発明者: 五十部 孝典 (**ISOBE, Takanori**); 〒1080075 東京都港区港南１丁目７番１号 ソニーグローバルマニュファクチャリング＆オペレーションズ株式会社内 Tokyo (JP). 樋渡 玄良(**HIWATARI, Harunaga**); 〒1080075 東京都港区港南１丁目７番１号 ソニーグローバルマニュファクチャリング＆オペレーションズ株式会社内 Tokyo (JP). ボグダノフ アンドレイ (**BOGDANOV, Andrey**); 2800 ビルディング３２４，ルーム２２１、コーゲンス、リュンビュー テクニカル ユニバーシティ オブ デンマーク内 Lyngby (DK).

(54) Title: ENCRYPTION DEVICE, ENCRYPTION METHOD, DECRYPTION DEVICE, AND DECRYPTION METHOD

(54) 発明の名称： 暗号化装置、暗号化方法、復号化装置、及び復号化方法



i	F(i)
0	X
1	Y
2	Z
⋮	⋮
2^n-1	W

i	F'(i)
0	W
1	Y
2	X
⋮	⋮
2^n-1	Z

(57) **Abstract:** [Problem] To provide an encryption calculation which is secure against side channel attacks and which can keep down processing loads. [Solution] This encryption device is provided with a data encryption unit that performs encryption with a white box model in which at least some of multiple rounding functions for sequentially performing encryption processing on input values are tabulated and in which the input and output values of the rounding functions can be recognized from the outside. Each of the aforementioned rounding functions has a tabulated encryption function

WO 2019/031025 A1

［続葉有］

(74) 代理人：亀谷 美明，外 (**KAMEYA, Yoshiaki** et al.)；〒1600004 東京都新宿区四谷３－１－３ 第一冨澤ビル はづき国際特許事務所四谷オフィス Tokyo (JP).

(81) 指定国(表示のない限り、全ての種類の国内保護が可能)：AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) 指定国(表示のない限り、全ての種類の広域保護が可能)：ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

添付公開書類：
— 国際調査報告（条約第21条(3)）

which encrypts input values in a black box model in which the input and output values can be recognized from the outside and intermediate values cannot be recognized from the outside, and the encryption functions are updated by means of random numbers.

(57) 要約：【課題】サイドチャネル攻撃に対して安全であり、且つ処理負荷を抑制することができる暗号化演算を提供する。【解決手段】本開示に係る暗号化装置は、入力値を順次に暗号化処理する複数のラウンド関数の少なくとも一部がテーブル化され、前記ラウンド関数の入出力値を外部から認識可能なホワイトボックスモデルにより暗号化するデータ暗号化部を備え、複数の前記ラウンド関数のそれぞれは、入出力値を外部から認識可能であり中間値は外部から認識できないブラックボックスモデルにおいて入力値を暗号化するテーブル化された暗号化関数を有し、前記暗号化関数が乱数により更新される。

明　細　書

発明の名称 :

暗号化装置、暗号化方法、復号化装置、及び復号化方法

技術分野

[0001]　本開示は、暗号化装置、暗号化方法、復号化装置、及び復号化方法に関する。

背景技術

[0002]　従来、下記の非特許文献１，２には、暗号化の中間値にマスキングと呼ばれる処理を施すことで、中間値と消費電力との依存関係を無くし、サイドチャネル攻撃に対する安全性を向上させることを想定した技術が記載されている。

[0003]　また、下記の特許文献３，４には、ホワイトボックスモデルで安全な暗号化方式が記載されている。

先行技術文献

非特許文献

[0004] 非特許文献1：J-S Coron, ”Higher Order Masking of Look-up Tables “ EUROCRYPTO2014

非特許文献2：T. S. Messerges, ”Securing the AES Finalists Against PowerAnalysis Attacks “, FSE 2000

非特許文献3：A. Bogdanov and T. Isobe, “Whitebox Cryptography Revisited:Space-hard Cipher” , ACM CCS 2015

非特許文献4：A. Bogdanov; T. Isobe; Elmar Tischhauser, “Towards PracticalWhitebox cryptography: Optimizing Efficiency and Space Hardness” , ASIACRYPT2016

発明の概要

発明が解決しようとする課題

[0005]　しかし、非特許文献１，２に記載された方法は、中間値と消費電力の依存

関係の一部しか無くすことができないため、事前に想定した特定の攻撃（ｄ−ｔｈ ｏｒｄｅｒ ａｔｔａｃｋ）に対しては証明可能な安全性を有するものの、特定の攻撃以外（ｄ＋１−ｔｈ ｏｒｄｅｒ ａｔｔａｃｋ）に対しての安全性を確保することができなかった。つまり、非特許文献１，２に記載された方法は、想定した攻撃レベルに応じて、性能を犠牲にした上で対策を施すことは可能であるが、想定を超えた攻撃に対しては耐性を確保することができなかった。

[0006] また、非特許文献１，２に記載された方法では、マスキングの処理を施すことにより処理負荷が非常に高くなるため、処理速度が遅くなり、実装性能が非常に悪くなる問題がある。具体的に、非特許文献１，２に記載された方法では、一般的な暗号化技術であるＡＥＳと比較すると、処理速度が数１０倍から数１０００倍程度になる問題がある。

[0007] 一方、非特許文献３，４に記載された方法では、攻撃者は暗号鍵を取得することはできないが、暗号化関数がテーブルで構成されるため、暗号鍵と等価なテーブルを攻撃者が取得した場合は、安全性を確保することができない問題がある。

[0008] そこで、サイドチャネル攻撃に対して安全であり、且つ処理負荷を抑制することができる暗号化演算が望まれていた。

課題を解決するための手段

[0009] 本開示によれば、入力値を順次に暗号化処理する複数のラウンド関数の少なくとも一部がテーブル化され、前記ラウンド関数の入出力値を外部から認識可能なホワイトボックスモデルにより暗号化するデータ暗号化部を備え、複数の前記ラウンド関数のそれぞれは、入出力値を外部から認識可能であり中間値は外部から認識できないブラックボックスモデルにおいて入力値を暗号化するテーブル化された暗号化関数を有し、前記暗号化関数が乱数により更新される、暗号化装置が提供される。

[0010] また、本開示によれば、入力値を順次に暗号化処理する複数のラウンド関数の少なくとも一部がテーブル化され、前記ラウンド関数の入出力値を外部

から認識可能なホワイトボックスモデルにより暗号化することを備え、複数の前記ラウンド関数のそれぞれは、入出力値を外部から認識可能であり中間値は外部から認識できないブラックボックスモデルにおいてテーブル化された暗号化関数により入力値を暗号化し、前記暗号化関数が乱数により更新される、暗号化方法が提供される。

[0011]　また、本開示によれば、入力値を順次に暗号化処理する複数のラウンド関数の少なくとも一部がテーブル化され、前記ラウンド関数の入出力値を外部から認識可能なホワイトボックスモデルにより暗号化する暗号化処理の逆演算により復号を行うデータ復号化部を備え、複数の前記ラウンド関数のそれぞれは、入出力値を外部から認識可能であり中間値は外部から認識できないブラックボックスモデルにおいてテーブル化された暗号化関数であって乱数により更新される前記暗号化関数により入力値を暗号化する、復号化装置が提供される。

[0012]　また、本開示によれば、入力値を順次に暗号化処理する複数のラウンド関数の少なくとも一部がテーブル化され、前記ラウンド関数の入出力値を外部から認識可能なホワイトボックスモデルにより暗号化する暗号化処理の逆演算により復号を行うことを備え、複数の前記ラウンド関数のそれぞれは、入出力値を外部から認識可能であり中間値は外部から認識できないブラックボックスモデルにおいてテーブル化された暗号化関数であって乱数により更新される前記暗号化関数によりにより入力値を暗号化する、復号化方法が提供される。

発明の効果

[0013]　以上説明したように本開示によれば、サイドチャネル攻撃に対して安全であり、且つ処理負荷を抑制することが可能となる。

　なお、上記の効果は必ずしも限定的なものではなく、上記の効果とともに、または上記の効果に代えて、本明細書に示されたいずれかの効果、または本明細書から把握され得る他の効果が奏されてもよい。

図面の簡単な説明

[0014] [図1]共通鍵ブロック暗号を示す模式図である。

[図2]暗号化を行うブロック（暗号関数Ｅ）の内部構成を示す模式図である。

[図3]Ｆｅｉｓｔｅｌ構造を示す模式図である。

[図4]ＳＰＮ構造を示す模式図である。

[図5]共通鍵ブロック暗号によって構成されるブラックボックスモデルを示す模式図である。

[図6]共通鍵ブロック暗号によって構成されるホワイトボックスモデルを示す模式図である。

[図7]本実施形態に係る暗号化技術の概要を示す模式図である。

[図8]具体例（Ｂ）、具体例（Ｃ）、具体例（Ｄ）、具体例（Ｅ）のそれぞれについいて、全体構成、Ｆ関数／Ｓ関数の種類、テーブルサイズ可変の可否、を示す模式図である。

[図9]暗号タイプに応じた処理を示すフローチャートである。

[図10]具体例（Ｂ）を示す模式図である。

[図11]Ｆ関数の構成を示す模式図である。

[図12]図１０において、ｎ＝ｎ’＝１２８、ｃ＝１、ｄ＝１６の場合の全体構成を示す模式図である。

[図13]図１２の例のＦ関数の構成を示している。

[図14]図１０において、ｎ＝１２８、ｃ＝１、ｄ＝８の場合の全体構成を示す模式図である。

[図15]図１０において、ｎ＝１２８、ｃ＝１、ｄ＝４の場合の全体構成を示す模式図である。

[図16]図１０において、ｎ＝１２８、ｃ＝３、ｄ＝１６の場合の全体構成を示す模式図である。

[図17]１つのラウンド内に２つＦ関数がある例であって、ｎ＝１２８、ｄ＝４の例を示す模式図である。

[図18]具体例（Ｃ）を示す模式図である。

[図19]図１８に示したＳ関数のそれぞれの構成を示す模式図である。

[図20]図１８において、ｎ＝１２８、ｄ＝８の場合を示す模式図である。

[図21]具体例（Ｄ）を示す模式図である。

[図22]具体例（Ｅ）を示す模式図である。

[図23]本実施形態に係る暗号化による安全性を説明するための模式図である。

[図24]本実施形態に係る暗号化による安全性を説明するための模式図である。

[図25]ブラックボックスモデル、ホワイトボックスモデルに対して、グレイボックスモデルの特徴を示す模式図である。

[図26]ホワイトボックスモデルで安全なブロック暗号から、グレイボックスモデルで安全なブロック暗号を生成する概要を示す模式図である。

[図27]テーブルの更新方法を示す模式図である。

[図28]図３に示したＦｅｉｓｔｅｌ構造の基本的な構成例において、乱数によりＦ関数を更新する例を示す模式図である。

[図29]図１５に示した具体的な構成例において、乱数によりＦ関数を更新する例を示す模式図である。

[図30]図２０に示したようなＳＰＮ構造による構成例において、乱数によりＳ関数を更新する例を示す模式図である。

[図31]著作権保護技術（ＤＲＭ：Digital Rights Management）への適用例を示す模式図である。

[図32]図３１をより詳細に示す模式図である。

[図33]ＮＦＣのエミュレーションを利用した支払システムへの適用例を示す模式図である。

[図34]図３３をより詳細に示す模式図である。

[図35]メモリリークに対しても安全な方式を示す模式図である。

[図36]サイドチャネル攻撃に対して安全な暗号化の例を示す模式図である。

発明を実施するための形態

[0015] 以下に添付図面を参照しながら、本開示の好適な実施の形態について詳細

に説明する。なお、本明細書及び図面において、実質的に同一の機能構成を有する構成要素については、同一の符号を付することにより重複説明を省略する。

[0016] なお、説明は以下の順序で行うものとする。

１．前提となる技術

２．本実施形態の概要

３．具体的構成例

３．１．具体例（Ｂ）

３．２．具体例（Ｃ）

３．３．具体例（Ｄ）

３．４．具体例（Ｅ）

４．ホワイトボックスモデルに係る暗号化による効果について

５．グレイボックスモデルで安全な構成

６．復号化のための構成例

７．既存技術との相違

８．本実施形態が適用されるアプリケーションの例

[0017] １．前提となる技術

暗号化と復号化に同じ鍵を用いる共通鍵ブロック暗号の技術が知られている。図１は、共通鍵ブロック暗号を示す模式図であって、ｋビットの鍵長に対応したｎビット共通鍵ブロック暗号アルゴリズムＥを示している。暗号化の際には、平文Ｐ（ｎビット）からｋビットの秘密鍵Ｋを用いて暗号関数Ｅにより暗号文Ｃ（ｎビット）が生成される。復号化の際には、暗号文Ｃ（ｎビット）からｋビットの秘密鍵Ｋを用いて復号関数Ｄ（$=E^{-1}$）により平文Ｐ（ｎビット）が生成される。このような共通鍵ブロック暗号では、例えば図１中に示すような通信路にデータが伝送される場合に、盗聴者（以下、攻撃者とも称する）に対して平文の秘匿性を実現できる。

[0018] 平文Ｐと暗号文Ｃのビット長をブロックサイズと称し、ここではｎで表す。ｎは任意の整数値を取りうるが、通常、ブロック暗号アルゴリズムごとに

予め１つに決められている。ブロック長がｎのブロック暗号のことをｎビットブロック暗号と称する場合がある。秘密鍵Ｋのビット長をｋで表し、鍵のビット長ｋは任意の整数値を取りうる。共通鍵ブロック暗号アルゴリズムは、１つまたは複数の鍵サイズに対応することになる。例えば、あるブロック暗号アルゴリズムＡはブロックサイズｎ＝１２８であり、ｋ＝１２８、またはｋ＝１９２、またはｋ＝２５６の鍵サイズに対応するという構成もあり得る。

[0019] 暗号化アルゴリズムＥに対応する復号アルゴリズムＤは、暗号化アルゴリズムＥの逆関数E^{-1}と定義でき、入力として暗号文Ｃと鍵Ｋを受け取り、平文Ｐを出力する。

[0020] 図２は、暗号化を行うブロック（暗号関数Ｅ）の内部構成を示す模式図である。暗号関数Ｅは、鍵スケジュール部１００とデータ暗号化部２００とから構成される。鍵スケジュール部１００は、秘密鍵Ｋを入力とし、ある定められたステップによりビット長を拡大してできた拡大鍵Ｋ’（ビット長ｋ’）を出力する。データ暗号化部２００は、平文Ｐを受け取り、鍵スケジュール部から拡大された拡大鍵Ｋ’を受け取ってデータの変換を行い、暗号文Ｃを出力する。データ暗号化部２００は、拡大鍵Ｋ’から得られるラウンド関数を繰り返し処理することで、暗号化を行う。

[0021] データ暗号化部２００は、処理単位であるラウンド関数に分割できるものとする。ラウンド関数は入力として２つのデータを受け取り、内部で処理を施したのち、１つのデータを出力する。入力データの一方は暗号化途中のｎビットデータであり、あるラウンドにおけるラウンド関数の出力が次のラウンド関数の入力として供給される。入力データの他方は鍵スケジュール部１００から出力された拡大鍵Ｋ’の一部のデータであり、この鍵データのことをラウンド鍵と称する。また、ラウンド関数の総数を総ラウンド数と称する。総ラウンド数は、暗号アルゴリズムごとに予め定められている値である。ここでは、総ラウンド数をＲで表す。データ暗号化部２００の入力側から１ラウンド目の入力データをX_1とし、ｉ番目のラウンド関数に入力されるデー

タをＸｉ、ラウンド鍵をＲＫｉとすると、データ暗号化部２００の構成は図２のように示される。

[0022] ブロック暗号アルゴリズムに応じてラウンド関数はさまざまな形態を取り得る。ラウンド関数はその暗号アルゴリズムが採用する構造によって分類できる。代表的な構造として、ここではＳＰＮ構造、Ｆｅｉｓｔｅｌ構造、拡張Ｆｅｉｓｔｅｌ構造を例示する。

[0023] 図３は、Ｆｅｉｓｔｅｌ構造を示す模式図である。また、図４は、ＳＰＮ構造を示す模式図である。図３に示すＦｅｉｓｔｅｌ構造の基本的な構成例では、各ラウンド関数において、ｎビットの入力データ$Ｘ_{ｉ}$が上位ｎ／２ビットと下位ｎ／２ビットに分割され、各ラインのデータのサイズはｎ／２ビットとなる。ここで、Ｆ関数には上位ｎ／２ビットが入力されて、ｎ／２ビットが出力される。この出力は下位ｎ／２ビットにそれぞれ排他的論理和される。その後、データの左右を入れ替えたものを出力データ$Ｘ_{ｉ＋１}$とする。Ｆ関数は非線形関数をもとに構成される。ＳＰＮ構造とは異なり、Ｆ関数は置換である必要はない。一般的に、Ｆ関数は、ブロック暗号から生成されることはなく、計算の軽い非線形演算で生成されるが、本実施形態では、Ｆ関数をブロック暗号から生成する。

[0024] 拡張Ｆｅｉｓｔｅｌ構造（一般化Ｆｅｉｓｔｅｌ構造）は、Ｆｅｉｓｔｅｌ構造ではデータ分割数が２であったものを３以上に分割する形に拡張したものである。分割数をｄとすると、分割数ｄによって様々な拡張Ｆｅｉｓｔｅｌ構造を定義することができる。Ｆ関数の入出力のサイズが相対的に小さくなるため、小型実装に向いている。また、各ラウンド関数において、複数のＦ関数を持つことができる。

[0025] 後述する図１７では、ｄ＝４であり、且つ１つのラウンド内に２つのＦ関数が並列に適用される場合の拡張Ｆｅｉｓｔｅｌ構造の一例を示している。この例では、第１Ｆ関数と第２Ｆ関数の各々において、$ＲＫ１_{ｉ}$と$ＲＫ２_{ｉ}$を鍵入力とする。また、後述する図１４では、ｄ＝８であり、且つ１つのラウンド内に１つのＦ関数が適用される場合の拡張Feistel構造の一例を示してい

る。この例では、Ｆ関数への入力サイズがｎ／８ビットであり、Ｆ関数からの出力サイズが７ｎ／８ビットであり、出力は７つのｎ／８ビットのデータに分割され、残りの７つの１６ビットデータに排他的論理和される。なお、ｎ＝１２８　ｂｉｔとする。

[0026]　また、図４に示すＳＰＮ構造の基本的な構成例では、ｎビットの入力データの全てに対して、ラウンド鍵との排他的論理和演算、非線形変換、線形変換処理などが適用される。非線形変換部をＳ層(Substitution-layer)、線形変換部をＰ層(Permutation-layer)と称し、それぞれは置換(全単射関数)である。各ラウンド関数において、ｎビットの入力データＸｉがｄ通りのデータに分割され、各ラインのデータのサイズはｎ／ｄ［ｂｉｔ］となる。ここで、非線形変換演算をＳ関数と定義し、各データ毎にｎ／ｄ［ｂｉｔ］の入出力の非線形変換演算Ｓ層(Substitution-layer)が実行される。その後、線形変換Ｐ層(Permutation-layer)としてｎビットの入出力線形変換Ｌが実行される。なお、線形変換演算をＬ関数と定義する。

[0027]　ブロック暗号の安全性モデルとして、ブラックボックスモデルとホワイトボックスモデルがある。図５は、共通鍵ブロック暗号によって構成されるブラックボックスモデルを示す模式図である。ブラックボックスモデルでは、秘密鍵を求めようとする攻撃者の能力が、ブロック暗号の入出力を認識して自由にコントロール可能であるが、攻撃者はブロック暗号の中間値は認識できない。つまり、ブラックボックスモデルは、攻撃者がブロック暗号アルゴリズムの入力、出力である平文Ｐ、暗号文Ｃにしかアクセスすることができない安全性モデルである。攻撃者による攻撃は、攻撃者が平文Ｐと暗号文Ｃのペアの値を知っているのみである既知平文暗号文攻撃と、これに加えて攻撃者が値自体を自由にコントロールできる選択平文暗号文攻撃に分けることができる。ブラックボックスモデルでは，暗号演算自体は安全に実行され、攻撃者が暗号の中間値を見たり、改ざんすることができないことを想定している。ハードウェアサポートなどを利用し、暗号演算の耐タンパ性が保障できている場合等が対応する。ブラックボックス用の暗号アルゴリズムの実装

方法をブラックボックス実装と称する。このようなブラックボックスモデルでは、攻撃者に秘密鍵を知られないように安全に設計することが可能である。ブラックボックスモデルにおいては、ブロック暗号は、秘密鍵Ｋを求めることが計算量的に困難であり（鍵回復攻撃耐性）、ブロック暗号と擬似ランダム鍵付置換を識別するのが計算量的に困難である（識別攻撃耐性）ように設計される。ブラックボックスモデルで安全なブロック暗号は、例えば、ＡＥＳ,ＣＬＥＦＩＡ,ＰＲＥＳＥＮＴ,Ｐｉｃｃｏｌｏなどの暗号化技術により実現可能である。

[0028] 図６は、共通鍵ブロック暗号によって構成されるホワイトボックスモデルを示す模式図である。ホワイトボックスモデルは、ブラックボックスモデルよりも強い攻撃者を想定した安全性モデルであり、攻撃者がブロック暗号アルゴリズムの入力、出力である平文Ｐ、暗号文Ｃのみならず、演算の中間値にも自由にアクセスできる。ホワイトボックスモデルでは、攻撃者はブロック暗号の入力である平文Ｐ、暗号文Ｃを自由にコントロールでき、更に攻撃者が演算中の任意の中間値を見ることや、改ざんできることを想定している。ハードウェアのサポートがないオールソフトウェアなどの、実装上の制約により、耐タンパが保障できないケースに対応している。また、バッファオーバーフロー等の実装上の脆弱性やマルウェア等により中間値が漏れてしまう場合にも対応する。ホワイトボックス用の暗号アルゴリズムの実装方法をホワイトボックス実装と称する。ホワイトボックス実装によれば、ソフトウェアのみでブロック暗号を構成することも可能である。

[0029] このように、ホワイトボックスモデルでは、攻撃者の能力が、ブロック暗号の入出力を認識して自由にコントロール可能であり、ブロック暗号の中間値を認識して自由にコントロール可能である。ホワイトボックスモデルにおいては、攻撃者が鍵Ｋを求めることは計算量的に困難であることが求められる。また、鍵Ｋを求める変わりにコード自体を直接用いて大きな鍵として用いる攻撃（ｃｏｄｅ ｌｉｆｔｉｎｇと呼ばれる）に対しての耐性も求められる。ブロック暗号の中間値を攻撃者が認識可能なホワイトボックスモデル

では、このような攻撃に対して定量的な安全性をもつ必要がある。

[0030]　２．本実施形態の概要

本実施形態では、上述したホワイトボックスモデルにおいて、信頼できない実行環境において、安全に暗号復号を行う技術、秘密鍵を守る技術を提案する。信頼できない環境として、秘密鍵を安全に保管できない場合、暗号演算の中間値を攻撃者が認識できる場合が挙げられる。

[0031]　図７は、本実施形態に係る暗号化技術の概要を示す模式図であって、基本となる構成例（Ａ）に係る暗号化装置を示している。ブロック暗号Ｅを複数のテーブル３００から構成し、各テーブルをブラックボックスモデルで安全なブロック暗号Ｅ’（内部ブロック暗号）として構成する。これにより、安全なブロック暗号Ｅを構成することができる。ホワイトボックス実装では、ブロック暗号Ｅ’からなる部品の一部、または全てをテーブル化して実装する。ブロック暗号Ｅ’のアルゴリズムはユーザが自由に選択可能とする。なお、暗号化装置は、ＣＰＵなどの中央演算処理装置と、これを機能させるためのプログラムから構成することができる。この場合に、そのプログラムは、暗号化装置が備えるメモリなどの記録媒体に格納されることができる。また、ブロック暗号を構成するテーブルは、暗号化装置が備える記録媒体に格納されることができる。

[0032]　このように、本実施形態の基本となる構成例（Ａ）では、ブラックボックスモデルで安全なブロック暗号Ｅ’を構成要素（部品）として、ホワイトボックスモデルで安全なブロック暗号Ｅを構成する。内部ブロック暗号Ｅ’のアルゴリズムは、ユーザが自由に選択でき、入力として受け取る。ホワイトボックス実装では、内部ブロック暗号Ｅ’ベースの関数を鍵に依存させ、一部もしくは全てがテーブルで実装される。つまり、鍵スケジュール部１００から出力された拡大鍵Ｋ’により内部ブロック暗号Ｅ’を生成してテーブル化する。テーブル化することにより、その都度暗号化演算が行われる場合と比較して、鍵の秘匿性を大幅に高めることできる。

[0033]　また、構成例（Ａ）の具体例（Ｂ）として、ブロック暗号ＥがＦｅｉｓｔ ｅｌ構造からなり、一種類の入出力サイズのＦ関数から構成され、Ｆ関数は内部ブロック暗号Ｅ’をもとに生成される。ここでＦ関数は、内部ブロック暗号Ｅ’の入力の一部を固定、出力の一部を破棄することによってＥ’から変換される。ホワイトボックス実装では、Ｆ関数すべてがテーブルで実装される。

[0034]　また、構成例（Ａ）の具体例（Ｃ）として、ブロック暗号ＥがＳＰＮ構造からなり、一種類の入出力サイズのＳ関数から構成され、Ｓ関数は内部ブロック暗号Ｅ’をもとに生成される。ここでＳ関数は、同じサイズの内部ブロック暗号から構成される。ホワイトボックス実装では、Ｓ関数すべてがテーブルで実装される。

[0035]　また、構成例（Ａ）の具体例（Ｄ）として、ブロック暗号Ｅが拡張Ｆｅｉｓｔｅｌ構造からなり、複数種類の入出力サイズのＦ関数から構成され、Ｆ関数は内部ブロック暗号Ｅ’をもとに生成される。ここで、Ｆ関数は、内部ブロック暗号の入力の一部を固定、出力の一部を破棄することによって生成される。ホワイトボックス実装では、一部若しくは全てがテーブル実装される。

[0036]　また、構成例（Ａ）の具体例（Ｅ）として、ブロック暗号ＥがＳＰＮ構造からなり、複数種類の入出力サイズのＳ関数から構成され、Ｓ関数は内部ブロック暗号Ｅ’をもとに生成する。ここで、Ｓ関数は、同じサイズの内部ブロック暗号から構成される。ホワイトボックス実装では、一部若しくは全てがテーブル実装される。

[0037]　図８は、具体例（Ｂ）、具体例（Ｃ）、具体例（Ｄ）、具体例（Ｅ）のそれぞれについいて、全体構成、Ｆ関数／Ｓ関数の種類、テーブルサイズ可変の可否、を示す模式図である。

[0038]　また、図９は、暗号タイプに応じた処理を示すフローチャートである。図９において、先ずステップＳ１０では、内部ブロック暗号Ｅ’に鍵Ｋを依存させ、鍵付関数$Ｅ’_{K}$を生成する。次のステップＳ１２では、暗号タイプを判定し、Ｆｅｉｓｔｅｌ構造の場合はステップＳ１４へ進む。ステップＳ１４

では、Ｆ関数をＥ’$_K$から生成する。次のステップＳ１６では、Ｆ関数をテーブル化する。次のステップＳ１８では、Ｆｅｉｓｔｅｌ構成でテーブルを接続し、暗号関数Ｅを生成する。

[0039] また、ステップＳ１２でＳＰＮ構造の場合はステップＳ２０へ進み、Ｓ関数をＥ’$_K$から生成する。次のステップＳ２２では、Ｓ関数をテーブル化する。次のステップＳ２４では、ＳＰＮ構成でテーブルを接続し、暗号関数Ｅを生成する。ステップＳ１８，Ｓ２４の後はステップＳ２６へ進み、テーブルベースの関数からコード生成を行う。これにより、ホワイトボックス暗号化コードが生成される。

[0040] ３．具体的構成例

以下では、具体例（Ｂ）、具体例（Ｃ）、具体例（Ｄ）、具体例（Ｅ）の構成例とその効果について詳細に説明する。ここで、内部ブロック暗号Ｅ’は、ｎ’ビットブロック暗号であり、ブラックボックスモデルにおいて安全であるものとする。

[0041] ３．１．具体例（Ｂ）

図１０は、具体例（Ｂ）を示す模式図であって、一般化Ｆｅｉｓｔｅｌ構造による構成例を示している。図１０に示す例では、ｎビットの入力データＸｉがｄ通りのデータに分割され、各ラインのデータのサイズはｎ／ｄビットとなる。ここで、Ｆ関数はｃ×ｎ／ｄ　ｂｉｔ入力、（ｄ－ｃ）×（ｎ／ｄ）（＝ｎ－（ｃ×ｎ／ｄ））［ｂｉｔ］出力で、ｃ通りのラインのデータが入力され、出力はｄ－ｃ通りのｎ／ｄ［ｂｉｔ］のデータに分割され、残りのｄ－ｃ通りのラインにそれぞれ排他的論理和される。Ｆ関数は内部ブロック暗号Ｅ’をもとに構成される。ここでＥ’のブロックサイズｎ’が、ｎ’＞（ｄ－ｃ）×（ｎ／ｄ）　かつ　ｎ’＞ｃ×（ｎ／ｄ）を満たすものとする（条件１）。図１０に示すように、ブロック暗号Ｅ’へ入力されたビットの値は、排他論理和により得られたビットの値よりも下位ビットとして出力される。

[0042] 図１１は、Ｆ関数の構成を示す模式図である。ｎ’ビットの内部ブロック

暗号Ｅ’から、ｃ×ｎ／ｄ［ｂｉｔ］入力、（ｄ－ｃ）×（ｎ／ｄ）［ｂｉｔ］出力のＦ関数の構成方法は以下の通りである。先ず、図１１に示すように、内部ブロック暗号Ｅ’の入力ｎ’［ｂｉｔ］のうち、任意のｎ’－（ｃ×ｎ／ｄ）［ｂｉｔ］を定数値（例えばall 0）に固定し、入力サイズをｃ×ｎ／ｄにする。次に、出力の任意の（ｃ×ｎ／ｄ）［ｂｉｔ］を破棄（ｄｉｓｒｅｇａｒｄ）することにより、出力サイズをｎ’－（ｃ×ｎ／ｄ）にする。このような、内部ブロック暗号Ｅ’に対する一部の入力ビット固定、出力の一部破棄により、条件１を満たす任意の内部ブロック暗号Ｅ’からＦ関数を構成する。テーブル化により、Ｆ関数はｎ’ビットの入出力に対応するテーブルから構成される。例えば、８ビットの入出力の場合、入力値（０～２５５）に対して出力値を対応付けしたテーブルが生成される。このテーブルに対し、一部の入力ビット固定、出力の一部破棄を行うことで、８ビット入力、１２０ビット出力などの入出力ビット数の調整を行うことができる。ここで、各ラウンドにおいてＦ関数を変更するため、ｎ’－（ｃ×ｎ／ｄ）ビットの出力にラウンド固有の定数を排他論理和（ＸＯＲ）する。例えば、ラウンド固有の定数は、例えばラウンド数とし、ラウンド数をＸＯＲする。ラウンド数４の場合は４をＸＯＲすることになる。但し、この排他論理和はテーブル参照後に行われるため、この演算自体はテーブルには含めない。これにより、一通りのＦ関数テーブルで、ラウンド毎に異なるＦ関数を表現することができる。従って、各ラウンド関数のＦ関数自体は共通に構成することも可能であり、テーブルを格納するメモリ領域を大幅に削減することができる。

[0043]　図１２～図１５は、具体的な構成例を示す模式図である。図１２はｎ＝ｎ’＝１２８、ｃ＝１、ｄ＝１６の場合の全体構成を示しており、図１３は、図１２の例のＦ関数の構成を示している。また、図１４は、ｎ＝１２８、ｃ＝１、ｄ＝８の場合、図１５はｎ＝１２８、ｃ＝１、ｄ＝４の場合、図１６はｎ＝１２８、ｃ＝３、ｄ＝１６の場合、をそれぞれ示している。

[0044]　図１７は、１つのラウンド内に２つＦ関数がある例であって、ｎ＝１２８

、ｄ＝４の例を示す模式図である。以上の全ての例において、Ｆ関数は、ホワイトボックス実装ではテーブル実装される。図１２、図１４、図１５、図１６の例において、テーブルサイズ（Ｆ関数のサイズ）は、それぞれ、約３．８４［ｂｙｔｅ］、９１８［Ｋｂｙｔｅ］、５１．５［Ｇｂｙｔｅ］、２１８［Ｍｂｙｔｅ］程度である。

[0045] ３．２．具体例（Ｃ）

図１８は、具体例（Ｃ）を示す模式図であって、ＳＰＮ構造による構成例を示している。図１８に示す例では、ｎビットの入力データＸｉがｄ通りのデータに分割され、各ラインのデータのサイズはｎ／ｄ［ｂｉｔ］である。ここで、各データ毎にｎ／ｄ［ｂｉｔ］の入出力のＳ関数による演算（非線形変換演算Ｓ層(Substitution-layer)）が実行される。その後、Ｌ関数による演算（線形変換Ｐ層(Permutation-layer)）としてｎ－ｂｉｔ入出力線形変換が実行される。ここで、Ｓ関数とＬ関数（入出力線形変換Ｌ）は全単射関数であり、Ｌ関数はラウンド定数演算を含む。Ｓ関数は、内部ブロック暗号Ｅ’をもとに構成されるが、Ｓ関数は全単射関数である必要があり、図１１のように内部ブロック暗号Ｅ’の入力ビット固定、出力の一部の破棄による変換では構成することができない。このため、ｎ／ｄ［ｂｉｔ］のブロック暗号を用いる必要がある。よって、内部ブロック暗号Ｅ’のブロックサイズｎ’の条件は、ｎ’＝ｎ／ｄとなる（条件２）。

[0046] 図１９は、図１８に示したＳ関数のそれぞれの構成を示す模式図である。図１９に示すように、Ｓ関数を構成する内部ブロック暗号Ｅ’の入出力のサイズは、共にｎ／ｄ［ｂｉｔ］である。従って、例えば、８ビットの入出力の場合、入力値（０～２５５）に対して出力値を対応付けしたテーブルが生成され、このテーブルによりＳ関数の演算が行われる。線形変換演算を行うＬ関数は、例えば正方行列から構成される。Ｓ関数の入出力が８ビットの場合、Ｓ関数からの８ビットの出力がＬ関数に入力され、８ビットの値に対して８×８のマトリクスの正方行列を乗算することで、８ビットの値がＬ関数から出力される。このように、Ｌ関数は、Ｓ関数からの出力値を拡散する機

能を有する。

[0047] 図２０は、具体的な構成例を示す模式図であって、ｎ＝１２８、ｄ＝８の場合を示している。Ｓ関数は、ホワイトボックス実装ではテーブル実装される。図２０のテーブルサイズは約２５６［ｂｙｔｅ］程度である。Ｓ関数の場合も、図１１に示したＦ関数の場合と同様に、各Ｓ関数を変更するため、Ｓ関数の出力にラウンド固有の定数をＸＯＲすることができる。これにより、Ｓ関数自体を共通にすることができるため、テーブルを格納するメモリ領域を大幅に削減することができる。

[0048] ３．３．具体例（Ｄ）

図２１は、具体例（Ｄ）を示す模式図であって、変形Ｆｅｉｓｔｅｌ構造による構成例を示している。図２１に示す例では、ｎビットの入力データがｄ通りのデータに分割され、各ラインのデータのサイズはｎ／ｄであり、サイズの異なる４種類のＦ関数から構成されている。初めのラウンドではｎ／ｄ［ｂｉｔ］入力、（ｎ−ｎ／ｄ）［ｂｉｔ］出力のＦ関数、２番目のラウンドでは２ｎ／ｄ［ｂｉｔ］入力、（ｎ−２ｎ／ｄ）［ｂｉｔ］出力のＦ関数、３番目のラウンドでは３ｎ／ｄ［ｂｉｔ］入力、（ｎ−３ｎ／ｄ）［ｂｉｔ］出力のＦ関数、４番目のラウンドでは４ｎ／ｄ［ｂｉｔ］入力、（ｎ−４ｎ／ｄ）［ｂｉｔ］出力のＦ関数が用いられる。この４ラウンドを基本として、任意ラウンド繰り返す。図１１で示した方法と同様に、任意のサイズのＦ関数は、内部ブロック暗号Ｅ’から生成され、出力にラウンド定数がＸＯＲされる。

[0049] ホワイトボックス実装では、ユーザの求めるコード(テーブルサイズ)に応じて、これらのうちの一部若しくは全てがテーブルで実装される。ｎ＝１２８、ｄ＝１６の場合は、それぞれのラウンドのＦ関数のテーブルサイズは、初めのラウンドでは約３．８４［ｂｙｔｅ］、２番目のラウンドでは９１８［Ｋｂｙｔｅ］、３番目のラウンドでは２１８［Ｍｂｙｔｅ］、４番目のラウンドでは５１．５［Ｇｂｙｔｅ］となる。ユーザの要求に応じて、どのＦ関数をテーブル実装するか選択することにより、全体のコードサイズを調整

可能である。例えば、４番目のラウンド関数はテーブル化せずに関数演算をその都度行うようにすれば、全体のコードサイズを抑えることができる。

[0050] ３．４．具体例（Ｅ）

図２２は、具体例（Ｅ）を示す模式図であって、変形ＳＰＮ構造による構成例を示している。図２２に示す例では、ｎビットの入力データがｄ通りのデータに分割され、各ラインのデータのサイズはｎ／ｄであり、サイズの異なる３種類のＳ関数から構成されている。各ラウンドのＳ層は、ｎ／ｄ［ｂｉｔ］の入出力、２ｎ／ｄ［ｂｉｔ］の入出力、４ｎ／ｄ［ｂｉｔ］の入出力、のＳ関数が用いられる。ホワイトボックス実装では、ユーザの求めるコード(テーブルサイズ)に応じて、これらの一部若しくは全てがテーブルで実装される。例えば。ｎ＝１２８、ｄ＝８で、８［ｂｉｔ］、１６［ｂｉｔ］、３２［ｂｉｔ］のデータが実装されている場合を考える。それぞれのテーブルサイズは、２５６［ｂｙｔｅ］、１３２［Ｋｂｙｔｅ］、１７．２［Ｇｂｙｔｅ］となる。ユーザの要求に応じて、どのＳ関数をテーブル実装するか選択することにより、全体のコードサイズを調整可能である。

[0051] 本実施形態によれば、ホワイトボックスモデルにおいて、key extractionの安全性は内部ブロック暗号Ｅ’のブラックボックスモデルでの鍵回復問題の安全性に帰着する。これは、ホワイトボックス実装において、内部ブロック暗号Ｅ’がテーブル実装されていることによるもので、攻撃者はホワイトモデルにおいても、テーブルの入出力にしかアクセスできない。これは、内部ブロック暗号Ｅ’のブラックボックスモデルとマッチする。内部状態（内部ブロック暗号Ｅ’）に信頼性の高い暗号(例えばＡＥＳ)を用ることで、ホワイトボックスモデルについても、内部ブロック暗号Ｅ’のブラックボックスモデルの鍵回復と同等の安全性を有することができる。

[0052] また、攻撃者は、鍵を知らない限りテーブルサイズを小さくすることはできない(Space-hardness)。内部ブロック暗号Ｅ’の鍵の情報を知らない限り、攻撃者はＥ’をテーブル演算以外で計算することはできない。このため、与えられたテーブルより小さいものに変換することはできない。これは、攻

撃者がｃｏｄｅ　ｌｉｆｔｉｎｇ攻撃する際に、大容量のデータが必要なことを意味している。データサイズに比例してコード抜き取りに必要な時間は増加するため、ｃｏｄｅ　ｌｉｆｔｉｎｇ作業を非常に時間のかかる作業にしている。更に、もしコード全体自体をとったとしても、そのサイズを圧縮することができず、コード配布の際に大容量のデータを送付する必要があり、配布のリスクを低減することが可能である。

[0053]　また、external encodingについても、External Encodingなしで安全性を保障可能である。

[0054]　更に、実装要求に応じた様々なサイズのテーブルを構成することが可能である。具体例（Ｂ）、具体例（Ｃ）によれば、分割数ｄの値を変更することにより、任意のテーブルサイズのアルゴリズムを構成することができる。また、具体例（Ｄ）、具体例（Ｅ）によれば、分割数ｄの値、又は使用するＦ関数、Ｓ関数のサイズを適切に複数選択することで、同じアルゴリズムで複数のテーブルサイズの実装が可能となる。

[0055]　また、ユーザは、内部ブロック暗号Ｅ’を自由に選択できる。内部ブロック暗号Ｅ’は、入出力サイズの条件(条件１、条件２)を満たす限り、自由に選択することが可能である。ブラックボックスで用いる場合、テーブル実装は不要で内部演算を直接演算できる。この場合、内部ブロック暗号Ｅ’を適切に選択することで、さまざまな実装ニーズに応えることが可能である。例えば、ＡＥＳを内部ブロック暗号Ｅ’として使用し、ＡＥＳ－ＮＩを用いることにより、ソフトウェアで非常に高速に実装でき、且つキャッシュタイミング攻撃に対しても安全に実装できる。また、ソフトウェアで軽量な暗号Ｐｉｃｃｏｌｏ,Ｐｒｉｄｅを用いることにより、ＲＡＭサイズ等の実装制約が大きい環境でも実装可能である。

[0056]　４．ホワイトボックスモデルに係る暗号化による効果について

図２３は、本実施形態に係るホワイトボックスモデル暗号化による安全性を説明するための模式図であって、図１１に示したＦｅｉｓｔｅｌ構造によるＦ関数をＡＥＳにより構成した例を示している。上述したように、ホワイ

トボックスモデルでは、攻撃者はテーブルの入出力にアクセス可能である。内部ブロック暗号Ｅ’のブラックボックスモデルと同じテーブルから鍵を求める問題（ホワイトボックスモデル）は、ＡＥＳの鍵回復攻撃(ブラックボックスモデル)と等しい。従って、本実施形態に係る暗号化によれば、ＡＥＳの鍵回復攻撃(ブラックボックスモデル)と同等の安全性を確保することが可能である。ホワイトボックスモデルにおいて、安全性は内部ブロック暗号Ｅ’のブラックボックスモデルでの鍵回復問題の安全性に帰着する。攻撃者は、鍵を知らない限りテーブルサイズを小さくすることはできない(Space-hardness)。

[0057]　図２４は、本実施形態に係る暗号化による安全性を説明するための模式図であって、攻撃者が攻撃する際に必要となるデータ量を示している。攻撃のためには、非常に多くのデータを入手しなければ秘密鍵Ｋを取得することはできない。具体的には、１２８　ｂｉｔ鍵と比較した場合、１０$^{4.4}$～１０$^{10.5}$倍のデータ量が必要である。また、攻撃者がデータを入手できたとしても、圧縮することができないので、不正に配布する際に大容量のデータであるため抑止力になる。

[0058]　また、本実施形態によれば、実装要求に応じた様々なサイズのテーブルを構成することが可能である。具体例(Ｂ),（Ｃ）の構成では、分割数ｄの数を変更することにより、任意のテーブルサイズのアルゴリズムを構成することができる。また、具体例（Ｄ），（Ｅ）の構成では、分割数ｄの数や使われるＦ関数、Ｓ関数のサイズを適切に複数選択することで、同じアルゴリズムで複数のテーブルサイズの実装が可能である。更に、テーブルの内部演算をユーザが自由に選択可能であり、ブラックボックス実装において最適なものを選択することが可能である。

[0059]　５．グレイボックスモデルで安全な構成

本実施形態では、以上説明したホワイトボックスモデルで安全なブロック暗号に対し、グレイボックスモデルで安全となるように構成を一部変更する。グレイボックスモデルでは、攻撃者は、ホワイトボックスモデルのように

演算の中間値を得ることはできないが、サイドチャネル情報を得ることができる。サイドチャネル情報として、電力の情報、演算のタイミングの情報、チップをプローブして得られる情報、暗号演算中に強い電磁波等を入力して誤動作させた際に得られる情報、等が挙げられる。これらのサイドチャネル情報は、演算の中間値そのものではないが、攻撃者により中間値を予測するために利用可能である。

[0060]　図２５は、ブラックボックスモデル、ホワイトボックスモデルに対して、グレイボックスモデルの特徴を示す模式図である。ブラックボックスモデル、グレイボックスモデル、ホワイトボックスモデルのいずれにおいても、攻撃者は入力値及び出力値を見ることができる。また、ブラックボックスモデルでは、攻撃者は暗号化の中間値を見ることができないが、ホワイトボックスモデルでは、攻撃者は暗号化の中間値をも見ることができる。また、グレイボックスモデルでは、上述のように攻撃者がサイドチャネル情報から中間値を予測することができる。従って、グレイボックスモデルでは、攻撃者は中間値を部分的に見ることができる。従って、図２５に示すように、攻撃者の能力は、ホワイトボックスモデルが最も高く、グレイボックスモデル、ブラックボックスモデルの順で低くなる。

[0061]　グレイボックスモデルでは、攻撃者の能力がホワイトボックスモデルよりも低く、グレイボックスモデルによれば、上述したブロック暗号は安全ではなく、サイドチャネル攻撃によりテーブルが復元されてしまう可能性がある。テーブルが復元されると、暗号鍵は判らないものの、暗号文が復元されてしまうことになる。このため、ホワイトボックスモデルの安全性を保ちつつ、グレイボックスモデルの安全性を保つことが求められる。

[0062]　このため、本実施形態では、上述したホワイトボックスモデルで安全なブロック暗号から、サイドチャネル攻撃に安全な暗号（グレイボックスモデルで安全な暗号）を生成する。図２６は、ホワイトボックスモデルで安全なブロック暗号から、グレイボックスモデルで安全なブロック暗号を生成する概要を示す模式図である。図２６に示すように、ホワイトボックスモデルでは

使用できないが、グレイボックスモデルでは使用できる乱数を用いてテーブルを動的に更新することで、グレイボックスモデルで安全なブロック暗号を生成する。これにより、テーブルの取得を目的とするグレイボックスモデルの攻撃者がテーブルを取得できないようにすることが可能である。

[0063]　図２７は、テーブルの更新方法を示す模式図である。図２７では、１つのＦ関数（テーブル）を示している。Ｆ関数へのデータの入力サイズはn_{in}とし、Ｆ関数からのデータの出力サイズはn_{out}とする。テーブルを更新する際には、n_{out}ビットの乱数r_{out}を生成し、n_{in}ビットの乱数r_{in}を選択する。入力に対し乱数r_{in}との排他的論理和をとり、その結果がＦ関数に入力される。また、Ｆ関数からの出力に対し、乱数r_{out}との排他的論理和をとり、その結果が出力される。これにより、Ｆ関数（Ｆ（ｉ））は、以下のようにＦ’（ｉ）に更新される。

$$F'(i) = F(i \hat{} r_{in}) \hat{} r_{out}$$

[0064]　ここで、Ｆ関数（テーブル）を更新したとしても、関数全体の機能を保持しておく必要がある。このため、乱数r_{out}がキャンセルされるように乱数r_{in}を選択する。図２８は、図３に示したＦｅｉｓｔｅｌ構造の基本的な構成例において、乱数によりＦ関数を更新する例を示す模式図である。更新後のＦｅｉｓｔｅｌ構造では、各Ｆ関数の入力と乱数r^{x}_{in}との排他的論理和が演算され、各Ｆ関数の出力と乱数r^{x}_{out}との排他的論理和が演算される。ここで、ラウンド数はｘ＋１となる。

[0065]　上述のように、各Ｆ関数の出力は、乱数r^{x}_{out}との排他的論理和が演算される。一方、入力側の乱数r_{in}は、図２８の左側に示す更新前のＦｅｉｓｔｅｌ構造と、右側に示す更新後のＦｅｉｓｔｅｌ構造を等価にするため、その値が調整される。前提として、平文Ｐを２つに分割して得られるデータP_{L}，P_{R}のうち、P_{L}は乱数r^{0}_{in}との排他的論理和が演算される。入力側の乱数のうち、r^{0}_{in}のみは調整により得られるものではなく、出力側の乱数r^{x}_{out}と同様に任意の乱数とされる。

[0066]　図２８において、入力側の乱数r^{1}_{in}，r^{2}_{in}，r^{3}_{in}，r^{4}_{in}，・・・，

r^{x}_{in}は、以下のように算出される。

$r^{1}_{in}=r^{0}_{out}$

$r^{2}_{in}=r^{0}_{In}\hat{}\ r^{1}_{out}$

$r^{3}_{in}=r^{1}_{in}\hat{}\ r^{2}_{out}$

$r^{4}_{in}=r^{2}_{in}\hat{}\ r^{3}_{out}$

・・・

$r^{x}_{in}=r^{x-2}_{in}\hat{}\ r^{x-1}_{out}$

[0067]　例えば、図２８の更新後のＦｅｉｓｔｅｌ構造において、r^{1}_{in}の値を調整する場合、各Ｆ関数への入力が更新前のＦｅｉｓｔｅｌ構造の各Ｆ関数への入力と一致する必要がある。更新後のＦｅｉｓｔｅｌ構造において、P_Lは乱数r^{0}_{in}との排他的論理和が演算されており、排他的論理和は２回繰り返すと元のデータに戻るため、Ｆ０関数への入力前に再びr^{0}_{in}との排他的論理和が演算されることで、Ｆ０関数への入力は更新前のＦｅｉｓｔｅｌ構造と一致する。同様に、更新後のＦｅｉｓｔｅｌ構造において、Ｆ１関数への入力については、Ｆ０関数からの出力と入力P_Rに対してr^{0}_{out}が排他的論理和されているため、Ｆ１関数への入力前に再びr^{0}_{out}との排他的論理和が演算されることで、Ｆ０関数への入力は更新前のＦｅｉｓｔｅｌ構造と一致する。従って、$r^{1}_{in}=r^{0}_{out}$となる。

[0068]　同様に、更新後のＦｅｉｓｔｅｌ構造において、Ｆ２関数への入力については、r^{2}_{in}が排他的論理和される以前のデータの流れ（図２８中に破線の矢印Ａ１で示す）では、更新前のＦｅｉｓｔｅｌ構造のデータの流れと比較すると、r^{0}_{In}とr^{1}_{out}によりデータがマスクされていることが判る。排他的論理和は２回繰り返すと元のデータに戻るため、$r^{2}_{in}=r^{0}_{In}\hat{}\ r^{1}_{out}$としてＦ２関数へ入力されるデータと$r^{2}_{in}$との排他的論理和をとることで、更新前のＦｅｉｓｔｅｌ構造のＦ２関数への入力と一致する。

[0069]　以上のようにして、上述のような乱数r^{1}_{in}，r^{2}_{in}，r^{3}_{in}，r^{4}_{in}，・・・，r^{x}_{in}を算出することができる。

[0070]　なお、図２８に示すように、更新後のＦｅｉｓｔｅｌ構造の最終的な出力

値は、C_{L}＾r^{4}_{in}，C_{R}＾r^{3}_{in}＾r^{4}_{out}となり、更新前のＦｅｉｓｔｅｌ構造の最終的な出力値C_{L}に対してマスクr^{4}_{in}がかかり、C_{R}に対してマスクr^{3}_{in}＾r^{4}_{out}がかかっている。このため、これらのマスクを除くことで、更新前のＦｅｉｓｔｅｌ構造と同じ出力値C_{L}，C_{R}を得ることができる。

[0071]　図２９は、図１５に示した具体的な構成例において、乱数によりＦ関数を更新する例を示す模式図である。更新方法は、図２８の例と同様である。図２９においても、Ｆ関数の出力側の乱数$r_{A}{}^{0}{}_{out}$，$r_{B}{}^{0}{}_{out}$，$r_{C}{}^{0}{}_{out}$，$r_{A}{}^{1}{}_{out}$，$r_{B}{}^{1}{}_{out}$，$r_{C}{}^{1}{}_{out}$，・・・に対して、入力側の乱数r^{1}_{in}，r^{2}_{in}，r^{3}_{in}，r^{4}_{in}，・・・，r^{x}_{in}を調整することで、乱数により更新する前の図１５の構成と等価にすることができる。

[0072]　例えば、図２９の構成において、Ｆ２関数への入力については、r^{2}_{in}が排他的論理和される以前のデータの流れ（図２９中に矢印Ａ２で示す）では、乱数により更新されていない図１５のデータの流れと比較すると、$r_{B}{}^{0}{}_{out}$と$r_{A}{}^{1}{}_{out}$によりデータがマスクされていることが判る。排他的論理和は２回繰り返すと元のデータに戻るため、$r^{2}_{in}=r_{A}{}^{1}{}_{out}$＾$r_{B}{}^{0}{}_{out}$としてＦ２関数へ入力されるデータと$r^{2}_{in}$との排他的論理和をとることで、更新前の図１５の構成のＦ２関数への入力と一致する。他のＦ関数への入力側についても同様の手法で乱数r^{1}_{in}，r^{2}_{in}，r^{3}_{in}，r^{4}_{in}，・・・，r^{x}_{in}を求めると、以下の通りとなる。

[0073] $r^{1}_{in}=r_{A}{}^{0}{}_{out}$

$r^{2}_{in}=r_{A}{}^{1}{}_{out}$＾$r_{B}{}^{0}{}_{out}$

$r^{3}_{in}=r_{A}{}^{2}{}_{out}$＾$r_{B}{}^{1}{}_{out}$＾$r_{c}{}^{0}{}_{out}$

$r^{4}_{in}=r_{A}{}^{3}{}_{out}$＾$r_{B}{}^{2}{}_{out}$＾$r_{c}{}^{1}{}_{out}$＾$r^{0}_{in}$

$r^{5}_{in}=r_{A}{}^{4}{}_{out}$＾$r_{B}{}^{3}{}_{out}$＾$r_{c}{}^{2}{}_{out}$＾$r^{1}_{in}$

$r^{x}_{in}=r_{A}{}^{x-1}{}_{out}$＾$r_{B}{}^{x-2}{}_{out}$＾$r_{c}{}^{x-3}{}_{out}$＾$r^{x-4}_{in}$

[0074]　図２９においても、乱数によりＦ関数を更新した後の最終的な出力値にはマスクがかかっているが、マスクを除くことで、更新前の構成と同じ出力値を得ることができる。

[0075]　図３０は、図２０に示したようなＳＰＮ構造による構成例において、乱数によりＳ関数を更新する例を示す模式図である。各Ｓ関数の前後には、乱数が排他的論理和される。この際、ｒ－１ラウンド目のＳ関数の出力側の乱数$r_{A}{}^{r-1}{}_{out}$，$r_{B}{}^{r-1}{}_{out}$，$r_{C}{}^{r-1}{}_{out}$，$r_{D}{}^{r-1}{}_{out}$とｒラウンド目のＳ関数の入力側の乱数$r_{A}{}^{r}{}_{in}$，$r_{B}{}^{r}{}_{in}$，$r_{C}{}^{r}{}_{in}$，$r_{D}{}^{r}{}_{in}$との間には、以下の関係が成立する。なお、図３０では、図２０のＬ関数を記号Ｍで表している。これにより、ｒ－１ラウンド目のＳ関数の出力側の乱数$r_{A}{}^{r-1}{}_{out}$，$r_{B}{}^{r-1}{}_{out}$，$r_{C}{}^{r-1}{}_{out}$，$r_{D}{}^{r-1}{}_{out}$をｒラウンド目のＳ関数の入力側の乱数$r_{A}{}^{r}{}_{in}$，$r_{B}{}^{r}{}_{in}$，$r_{C}{}^{r}{}_{in}$，$r_{D}{}^{r}{}_{in}$でキャンセルすることができる。

[0076] [数1]

$$\begin{pmatrix} r_{A}{}^{r}{}_{in} \\ r_{B}{}^{r}{}_{in} \\ r_{C}{}^{r}{}_{in} \\ r_{D}{}^{r}{}_{in} \end{pmatrix} = M \cdot \begin{pmatrix} r_{A}{}^{r-1}{}_{out} \\ r_{B}{}^{r-1}{}_{out} \\ r_{C}{}^{r-1}{}_{out} \\ r_{D}{}^{r-1}{}_{out} \end{pmatrix}$$

[0077]　次に、上述した手法により更新された関数（テーブル）の安全性について説明する。図２７に示すＦ関数において、更新前のＦ関数は疑似ランダム関数であるものとし、ｉランド目の$r^{i}{}_{in}$，$r^{i}{}_{out}$は乱数であるものとする。なお、擬似ランダム関数とは、攻撃者が真のランダム関数と疑似ランダム関数の双方の入出力にアクセスできたとしても、攻撃者がどちらの関数かを識別できない関数をいう。

[0078]　図２７において、更新されたＦ’（ｉ）も疑似ランダム関数である。また、更新されたＦ’（ｉ）から$r^{i}{}_{in}$，$r^{i}{}_{out}$の情報が漏れることはない。そして、更新されたＦ’（ｉ）から更新前のＦ関数の情報が漏れることはない。従って、更新毎にランダムな関数（テーブル）を生成することができる。

[0079]　次に、更新のタイミングについて説明する。乱数による関数の更新は、データが入力される毎に行っても良いか、処理負荷を下げるためには、攻撃者

がテーブルを回復できない範囲で更新の頻度を下げることが望ましい。

[0080] 　ここで、Ｘ通りのテーブルを攻撃者が回復するためには、少なくともＸ回の暗号演算が必要である。また、ホワイトボックスモデルで安全なブロック暗号は、テーブルエントリの１／４以下を攻撃者に取得されても安全である。従って、テーブルのエントリ数を2^{n}としたときに、2^{n-2}の実行毎にテーブルの更新を行えば良い。これにより、テーブルの１／４以下が攻撃者に取得される可能性はあるが、安全性を確実に保つことができる。

[0081] 　６．復号化のための構成例

上述したように、暗号化アルゴリズムＥに対応する復号アルゴリズムＤは、暗号化アルゴリズムＥの逆関数E^{-1}と定義でき、入力として暗号文Ｃと鍵Ｋを受け取り、平文Ｐを出力する。復号アルゴリズムＤにおいても、ブラックボックス実装によりテーブルを構成することで、ブラックボックスモデルと同等の安全性を確保することが可能である。

[0082] 　７．既存技術との相違

以下では、本実施形態に係る技術と、前述した非特許文献１，２に記載された方法（既存技術１とする）、非特許文献３，４に記載された方法（既存技術２とする）との相違について説明する。

[0083] 　既存技術1は、中間値と消費電力の依存関係の一部しか無くすことができないため、事前に想定した１st ａｎｄ ２nd ｏｒｄｅｒ ａｔｔａｃｋなどの特定の攻撃（ｄ－ｔｈ ｏｒｄｅｒ ａｔｔａｃｋ）に対しては証明可能な安全性を有するものの、３rd ｏｒｄｅｒ ａｔｔａｃｋなどの特定の攻撃以外（ｄ＋１－ｔｈ ｏｒｄｅｒ ａｔｔａｃｋ）に対しての安全性を確保することができない。つまり、既存技術１は、限定された攻撃に対しての対策技術に過ぎない。

[0084] 　また、既存技術１では、マスキングの処理を施すことにより処理負荷が非常に高くなるため、処理速度が遅くなり、実装性能が非常に悪くなる問題がある。既存技術１では、一般的な暗号化技術であるＡＥＳと比較すると、処理速度が数１０倍から数１０００倍程度になる問題がある。

[0085] また、既存技術２では、攻撃者が暗号鍵を取得することはできないものの、攻撃者が暗号鍵と実質的に等価であるテーブルの情報を取得した場合は安全性を保つことができない。

[0086] 一方、本実施形態に係る乱数により暗号化関数を更新する手法では、サイドチャネル攻撃を含むあらゆる攻撃に対して耐性を確保することができる。また、処理負荷に関しても、基本的に暗号化関数に乱数を付加することで構成できるため、既存技術１よりも大幅に低い処理負荷で実現が可能である。

[0087] ８．本実施形態が適用されるアプリケーションの例

本実施形態に係る技術は、図１に示したような通信路におけるデータの秘匿性を実現する他、様々なアプリケーションに適用することができる。以下では、幾つかのアプリケーションの例を説明する。

[0088] 図３１は、著作権保護技術（ＤＲＭ：Digital Rights Management）への適用例を示す模式図である。図３１に示すように、クラウド上のコンテンツサーバ４００で暗号化を行い、コンテンツ（暗号文Ｃ）がコンテンツサーバ４００からユーザデバイス４１０へ配信される。ユーザデバイス４１０は、パーソナルコンピュータ（ＰＣ）、スマートフォンなどの電子機器である。コンテンツ（暗号文Ｃ）はユーザデバイス４１０において復号される。

[0089] 図３２は、図３１をより詳細に示す模式図である。コンテンツサーバ４００は、ホワイトボックス暗号化関数により映画、音楽などのコンテンツを暗号化する。また、コンテンツサーバ４００では、ライセンス生成器４０２によりライセンスを生成し、暗号化されたコンテンツと共にユーザデバイス４１０へ送る。ユーザデバイス４１０は、送られたライセンスをライセンス検証器４１２により検証し、ライセンスの検証に成功した場合は、ホワイトボックス復号関数により、暗号化されたコンテンツを復号する。

[0090] 図３１及び図３２に示したような著作権保護技術では、ユーザデバイス４１０でコンテンツを復号する必要がある。この場合、仮に鍵Ｋが露呈した場合、コンテンツの不正配布に繋がる。つまり、暗号化が安全でない環境では、ユーザデバイス４１０が信頼できない環境となる。本実施形態によれば、

悪意のあるユーザがコンテンツの秘密鍵Ｋを取得することを、ホワイトボックス暗号化技術により確実に防止することが可能である。

[0091] 図３３は、ＮＦＣのエミュレーションを利用した支払システムへの適用例を示す模式図である。図３３に示すように、このシステムでは、ＮＦＣの読取装置４２０にユーザデバイス４３０を近づけてエミュレーションを行う。ユーザデバイス４３０は、ホストＣＰＵ４３２、ＮＦＣコントローラ４３４、セキュアエレメント４３６を有する。

[0092] 図３４は、図３３をより詳細に示す模式図である。クラウド上のサーバ４４０は、ユーザの証明用の情報（Ｃｒｅｄｅｎｔｉａｌ ｉｎｆｏｒｍａｔｉｏｎ）と、支払情報（Ｐａｙｍｅｎｔ ｉｎｆｏｒｍａｔｉｏｎ）を有する。ユーザデバイス４３０は、モバイル機器などの電子機器であり、サーバ４４０と暗号化通信を行い、証明用の情報をやり取りする。また、ユーザデバイス４３０は、読取装置４２０と暗号化通信を行い、証明用の情報をやり取りする。暗号化通信では、本実施形態に係るホワイトボックス暗号化により暗号化が行われる。このため、ユーザデバイス４３０は、ホワイトボックス暗号関数、復号関数を備えている。ホワイトボックス暗号化により暗号化を行うことで、支払いに関する証明のデータを守ることができ、ユーザデバイス４３０がセキュアエレメント４３６を備えていなくてもＮＦＣのエミュレーションが可能となる。

[0093] 図３５は、メモリリークに対しても安全な方式を示す模式図である。このシステムでは、ソフトウェアの脆弱性(buffer overflow, heart bleed)やマルウェアにより、メモリが攻撃者にリークした場合も、安全性を保障する。マルウェアやメモリリークの脆弱性のあるデバイス４４５において、ホワイトボックス暗号化方式が有するＳｐａｃｅ ｈａｒｄｎｅｓｓの性質により、数キロバイト、数ギガバイト以上のデータが漏れない限り安全性は低下しない。図３５の例において、コードサイズをＴとすると、Ｔ／４以上のデータが漏れない限り安全性は低下しない。なお、Ｓｐａｃｅ ｈａｒｄｎｅｓｓとは、一定以上のサイズのメモリが漏れない限り暗号の安全性を保障でき

る技術である。この方法は、内部ネットワークから外部ネットワークからの通信量を制限している場合に特に有効である。

[0094]　図３６は、サイドチャネル攻撃に対して安全な暗号化の例を示す模式図である。ホワイトボックス暗号化方式は、通常はソフトウェア用であるが、Reconfigurable Hardware (FPGA)で実装することにより、ハードウェアでサイドチャネルに安全な暗号方式として使うことができる。例えば図３６に示すＩＣカード４５０など、ハードウェアでサイドチャネル攻撃を受ける可能性のあるデバイスにおいて、特に有効である。

[0095]　以上、添付図面を参照しながら本開示の好適な実施形態について詳細に説明したが、本開示の技術的範囲はかかる例に限定されない。本開示の技術分野における通常の知識を有する者であれば、特許請求の範囲に記載された技術的思想の範疇内において、各種の変更例または修正例に想到し得ることは明らかであり、これらについても、当然に本開示の技術的範囲に属するものと了解される。

[0096]　また、本明細書に記載された効果は、あくまで説明的または例示的なものであって限定的ではない。つまり、本開示に係る技術は、上記の効果とともに、または上記の効果に代えて、本明細書の記載から当業者には明らかな他の効果を奏しうる。

[0097]　なお、以下のような構成も本開示の技術的範囲に属する。

（１）　入力値を順次に暗号化処理する複数のラウンド関数の少なくとも一部がテーブル化され、前記ラウンド関数の入出力値を外部から認識可能なホワイトボックスモデルにより暗号化するデータ暗号化部を備え、

複数の前記ラウンド関数のそれぞれは、入出力値を外部から認識可能であり中間値は外部から認識できないブラックボックスモデルにおいて入力値を暗号化するテーブル化された暗号化関数を有し、

前記暗号化関数が乱数により更新される、暗号化装置。

（２）　前記暗号化関数の入力値に第１の係数が排他論理和され、前記暗号化関数の出力値に第２の係数が排他論理和され、少なくとも前記第２の係数

が乱数である、前記（１）に記載の暗号化装置。

（３）　前記第１の係数は、前記出力値に前記第２の係数が排他論理和されたことによる前記データ暗号化部の変化を打ち消す値に調整される、前記（２）に記載の暗号化装置。

（４）　任意のラウンドの前記暗号化関数の前記第１の係数は、当該任意のラウンドより前のラウンドの前記出力値に前記第２の係数が排他論理和されたことによる当該任意のラウンドの前記暗号化関数への前記入力値の変化を打ち消す値に調整される、前記（３）に記載の暗号化装置。

（５）　前記暗号化関数には、前記ラウンド関数へ入力されるビットのうちの一部が入力され、

前記暗号化関数は、前記暗号化関数へ入力可能なビットの一部を固定値とし、前記暗号化関数の出力値の一部を破棄することで、前記暗号化関数へ入力可能なビット数から前記暗号化関数へ入力されたビット数の差分に相当するビット数の出力値を出力する、前記（１）に記載の暗号化装置。

（６）　前記ラウンド関数は、前記ラウンド関数へ入力されるビットのうち前記暗号化関数に入力されなかったビットと、前記暗号化関数からの前記出力値のビットの排他論理和を演算する、前記（５）に記載の暗号化装置。

（７）　前記ラウンド関数は、前記暗号化関数へ入力されたビットの値と前記排他論理和により得られたビットの値を出力する、前記（６）に記載の暗号化装置。

（８）　前記ラウンド関数は、前記暗号化関数へ入力されたビットの値を前記排他論理和により得られたビットの値よりも下位ビットとして出力する、前記（７）に記載の暗号化装置。

（９）　前記ラウンド関数の出力と予め定められた所定値との排他論理和を演算し、得られた値を次のラウンド関数への入力又は前記データ暗号化部の出力とする、前記（５）～（８）のいずれかに記載の暗号化装置。

（１０）　１の前記ラウンド関数が複数の前記暗号化関数を有する、前記（１）～（９）のいずれかに記載の暗号化装置。

（１１）　複数の前記ラウンド関数において、後段のラウンド関数ほど前記暗号化関数に大きなビットの入力値が入力される、前記（５）～（９）のいずれかに記載の暗号化装置。

（１２）　１の前記ラウンド関数が複数の前記暗号化関数を有し、

前記ラウンド関数へ入力されるビットが分割されて複数の前記暗号化関数へ入力され、

複数の前記暗号化関数は非線形演算を行い、

前記ラウンド関数は、複数の前記暗号化関数による前記非線形演算の結果を線形変換演算して出力する、前記（１）～（１１）のいずれかに記載の暗号化装置。

（１３）　複数の前記暗号化関数のそれぞれにおいて、入力されるビット数と出力されるビット数が同一である、前記（１２）に記載の暗号化装置。

（１４）　複数の前記暗号化関数のそれぞれに入力されるビット数が異なる、前記（９）又は（１３）に記載の暗号化装置。

（１５）　前記暗号化関数は、前記データ暗号化部に対応する秘密鍵から生成される拡張鍵によって暗号化を行う、前記（１）～（１４）のいずれかに記載の暗号化装置。

（１６）　　入力値を順次に暗号化処理する複数のラウンド関数の少なくとも一部がテーブル化され、前記ラウンド関数の入出力値を外部から認識可能なホワイトボックスモデルにより暗号化することを備え、

複数の前記ラウンド関数のそれぞれは、入出力値を外部から認識可能であり中間値は外部から認識できないブラックボックスモデルにおいてテーブル化された暗号化関数により入力値を暗号化し、

前記暗号化関数が乱数により更新される、暗号化方法。

（１７）　入力値を順次に暗号化処理する複数のラウンド関数の少なくとも一部がテーブル化され、前記ラウンド関数の入出力値を外部から認識可能なホワイトボックスモデルにより暗号化する暗号化処理の逆演算により復号を行うデータ復号化部を備え、

　複数の前記ラウンド関数のそれぞれは、入出力値を外部から認識可能であり中間値は外部から認識できないブラックボックスモデルにおいてテーブル化された暗号化関数であって乱数により更新される前記暗号化関数により入力値を暗号化する、復号化装置。

（１８）　入力値を順次に暗号化処理する複数のラウンド関数の少なくとも一部がテーブル化され、前記ラウンド関数の入出力値を外部から認識可能なホワイトボックスモデルにより暗号化する暗号化処理の逆演算により復号を行うことを備え、

　複数の前記ラウンド関数のそれぞれは、入出力値を外部から認識可能であり中間値は外部から認識できないブラックボックスモデルにおいてテーブル化された暗号化関数であって乱数により更新される前記暗号化関数によりにより入力値を暗号化する、復号化方法。

符号の説明

[0098]　２００　データ暗号化部

３００　テーブル

請求の範囲

[請求項1] 入力値を順次に暗号化処理する複数のラウンド関数の少なくとも一部がテーブル化され、前記ラウンド関数の入出力値を外部から認識可能なホワイトボックスモデルにより暗号化するデータ暗号化部を備え、

複数の前記ラウンド関数のそれぞれは、入出力値を外部から認識可能であり中間値は外部から認識できないブラックボックスモデルにおいて入力値を暗号化するテーブル化された暗号化関数を有し、

前記暗号化関数が乱数により更新される、暗号化装置。

[請求項2] 前記暗号化関数の入力値に第１の係数が排他論理和され、前記暗号化関数の出力値に第２の係数が排他論理和され、少なくとも前記第２の係数が乱数である、請求項１に記載の暗号化装置。

[請求項3] 前記第１の係数は、前記出力値に前記第２の係数が排他論理和されたことによる前記データ暗号化部の変化を打ち消す値に調整される、請求項２に記載の暗号化装置。

[請求項4] 任意のラウンドの前記暗号化関数の前記第１の係数は、当該任意のラウンドより前のラウンドの前記出力値に前記第２の係数が排他論理和されたことによる当該任意のラウンドの前記暗号化関数への前記入力値の変化を打ち消す値に調整される、請求項３に記載の暗号化装置。

[請求項5] 前記暗号化関数には、前記ラウンド関数へ入力されるビットのうちの一部が入力され、

前記暗号化関数は、前記暗号化関数へ入力可能なビットの一部を固定値とし、前記暗号化関数の出力値の一部を破棄することで、前記暗号化関数へ入力可能なビット数から前記暗号化関数へ入力されたビット数の差分に相当するビット数の出力値を出力する、請求項１に記載の暗号化装置。

[請求項6] 前記ラウンド関数は、前記ラウンド関数へ入力されるビットのうち

前記暗号化関数に入力されなかったビットと、前記暗号化関数からの前記出力値のビットの排他論理和を演算する、請求項５に記載の暗号化装置。

[請求項7] 前記ラウンド関数は、前記暗号化関数へ入力されたビットの値と前記排他論理和により得られたビットの値を出力する、請求項６に記載の暗号化装置。

[請求項8] 前記ラウンド関数は、前記暗号化関数へ入力されたビットの値を前記排他論理和により得られたビットの値よりも下位ビットとして出力する、請求項７に記載の暗号化装置。

[請求項9] 前記ラウンド関数の出力と予め定められた所定値との排他論理和を演算し、得られた値を次のラウンド関数への入力又は前記データ暗号化部の出力とする、請求項１に記載の暗号化装置。

[請求項10] １の前記ラウンド関数が複数の前記暗号化関数を有する、請求項１に記載の暗号化装置。

[請求項11] 複数の前記ラウンド関数において、後段のラウンド関数ほど前記暗号化関数に大きなビットの入力値が入力される、請求項５に記載の暗号化装置。

[請求項12] １の前記ラウンド関数が複数の前記暗号化関数を有し、

前記ラウンド関数へ入力されるビットが分割されて複数の前記暗号化関数へ入力され、

複数の前記暗号化関数は非線形演算を行い、

前記ラウンド関数は、複数の前記暗号化関数による前記非線形演算の結果を線形変換演算して出力する、請求項１に記載の暗号化装置。

[請求項13] 複数の前記暗号化関数のそれぞれにおいて、入力されるビット数と出力されるビット数が同一である、請求項１２に記載の暗号化装置。

[請求項14] 複数の前記暗号化関数のそれぞれに入力されるビット数が異なる、請求項１２に記載の暗号化装置。

[請求項15] 前記暗号化関数は、前記データ暗号化部に対応する秘密鍵から生成

される拡張鍵によって暗号化を行う、請求項１に記載の暗号化装置。

[請求項16] 入力値を順次に暗号化処理する複数のラウンド関数の少なくとも一部がテーブル化され、前記ラウンド関数の入出力値を外部から認識可能なホワイトボックスモデルにより暗号化することを備え、

複数の前記ラウンド関数のそれぞれは、入出力値を外部から認識可能であり中間値は外部から認識できないブラックボックスモデルにおいてテーブル化された暗号化関数により入力値を暗号化し、

前記暗号化関数が乱数により更新される、暗号化方法。

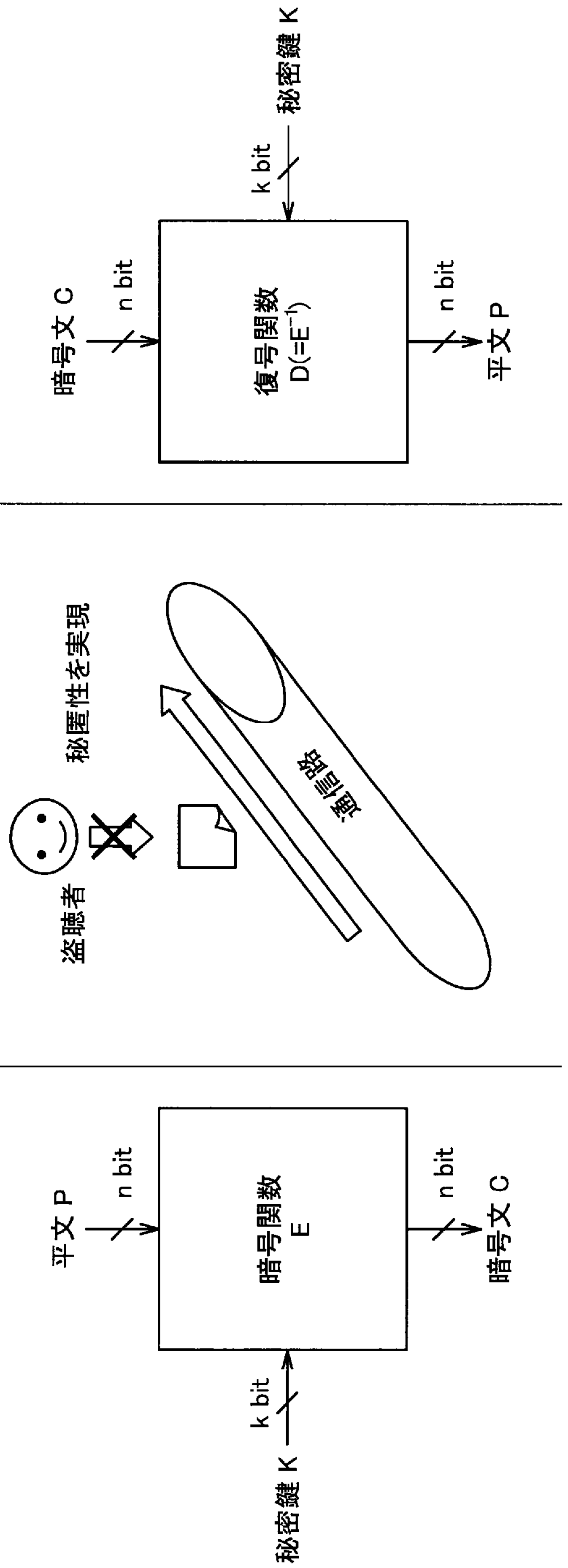
[請求項17] 入力値を順次に暗号化処理する複数のラウンド関数の少なくとも一部がテーブル化され、前記ラウンド関数の入出力値を外部から認識可能なホワイトボックスモデルにより暗号化する暗号化処理の逆演算により復号を行うデータ復号化部を備え、

複数の前記ラウンド関数のそれぞれは、入出力値を外部から認識可能であり中間値は外部から認識できないブラックボックスモデルにおいてテーブル化された暗号化関数であって乱数により更新される前記暗号化関数により入力値を暗号化する、復号化装置。

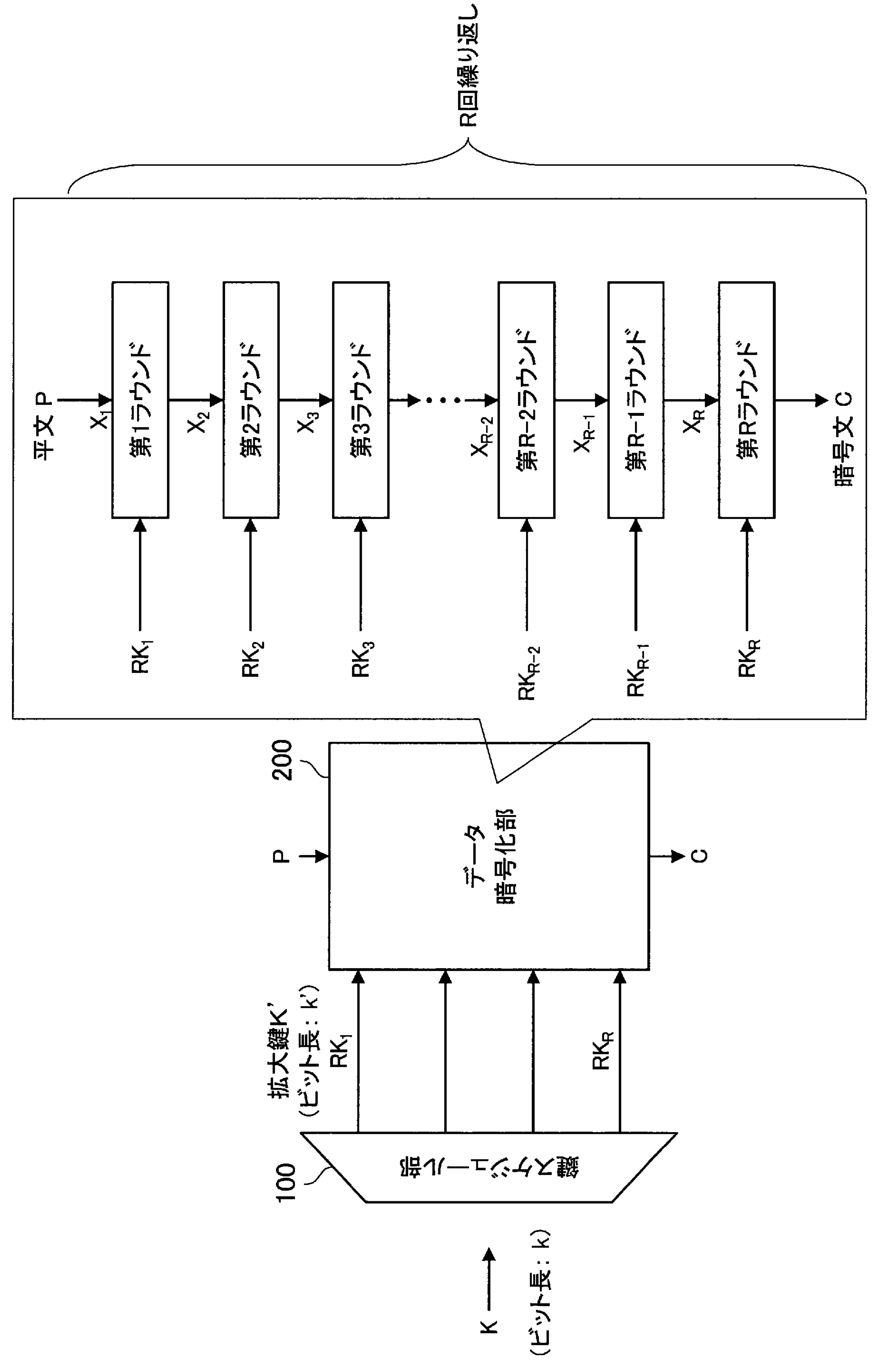
[請求項18] 入力値を順次に暗号化処理する複数のラウンド関数の少なくとも一部がテーブル化され、前記ラウンド関数の入出力値を外部から認識可能なホワイトボックスモデルにより暗号化する暗号化処理の逆演算により復号を行うことを備え、

複数の前記ラウンド関数のそれぞれは、入出力値を外部から認識可能であり中間値は外部から認識できないブラックボックスモデルにおいてテーブル化された暗号化関数であって乱数により更新される前記暗号化関数によりにより入力値を暗号化する、復号化方法。

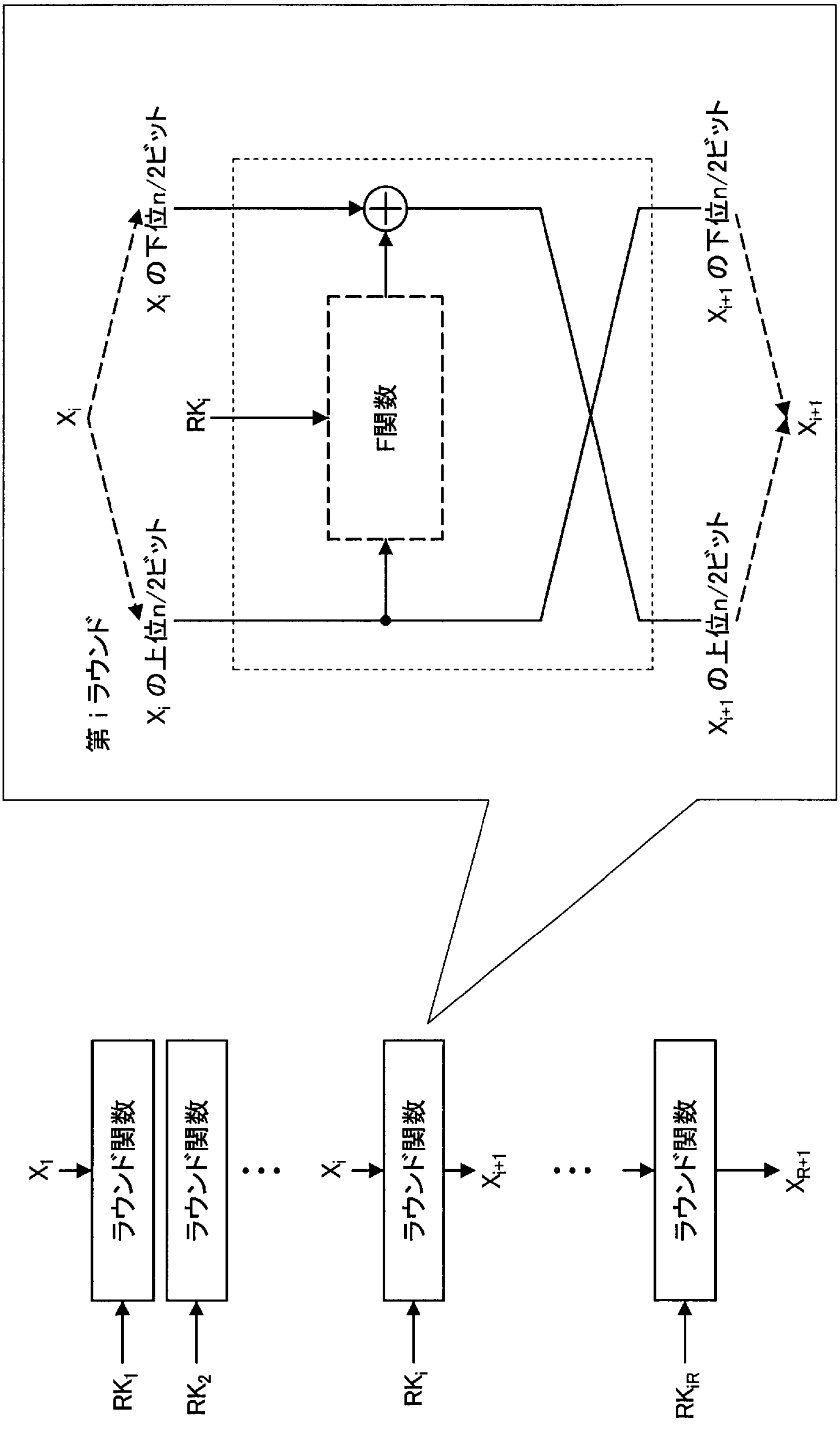
[図1]


秘密鍵 K
k bit
平文 P
n bit
暗号関数
E
n bit
暗号文 C
盗聴者
秘匿性を実現
通信路
暗号文 C
n bit
復号関数
D(=E^{-1})
n bit
平文 P
秘密鍵 K
k bit

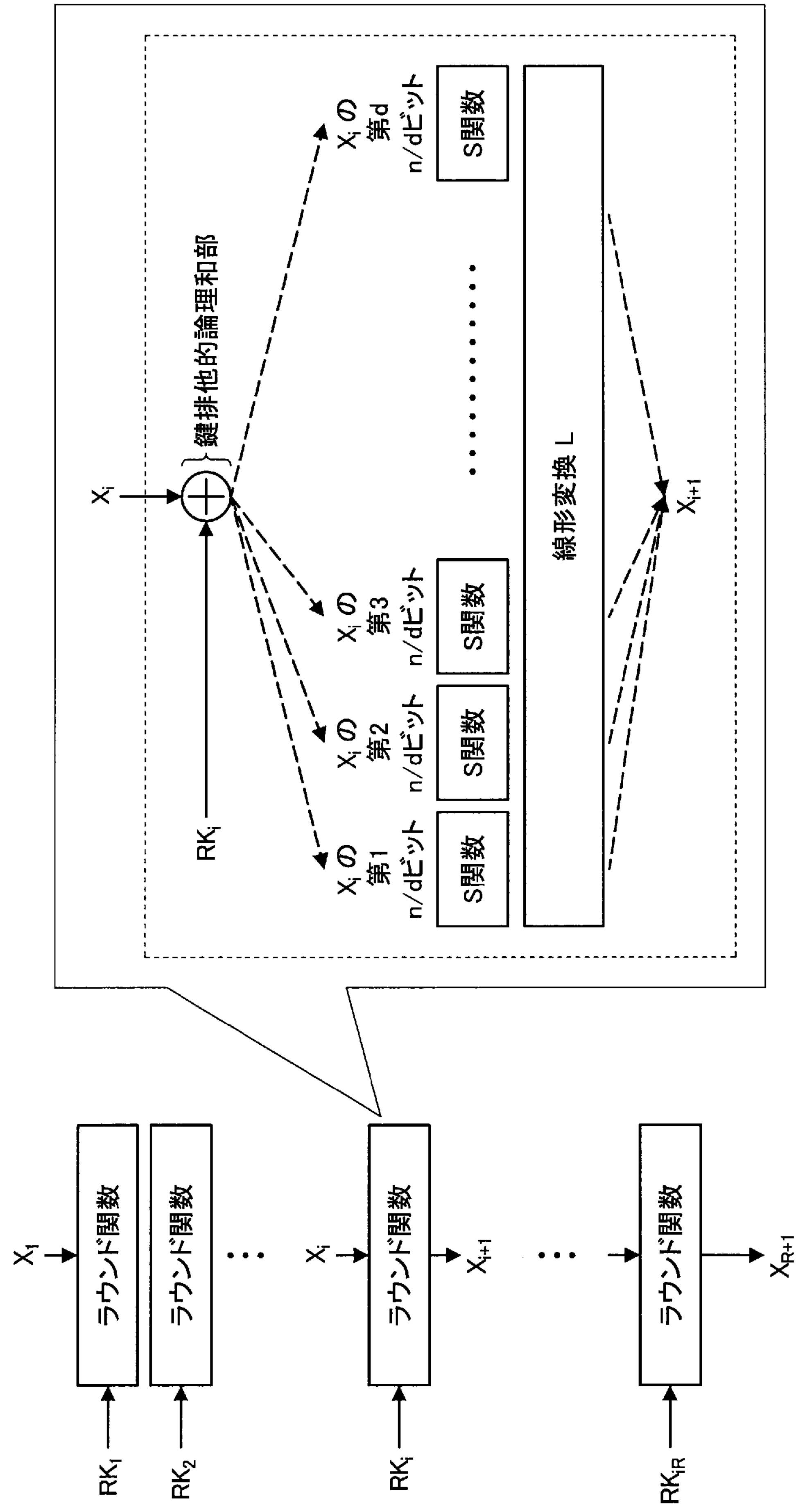
[図2]


R回繰り返し
平文 P
X_1
第1ラウンド
X_2
第2ラウンド
X_3
第3ラウンド
X_{R-2}
第R−2ラウンド
X_{R-1}
第R−1ラウンド
X_R
第Rラウンド
暗号文 C
RK_1
RK_2
RK_3
RK_{R-2}
RK_{R-1}
RK_R
200
P
データ暗号化部
C
拡大鍵K'（ビット長：k'）
RK_1
RK_R
100
鍵スケジュール部
K
（ビット長：k）

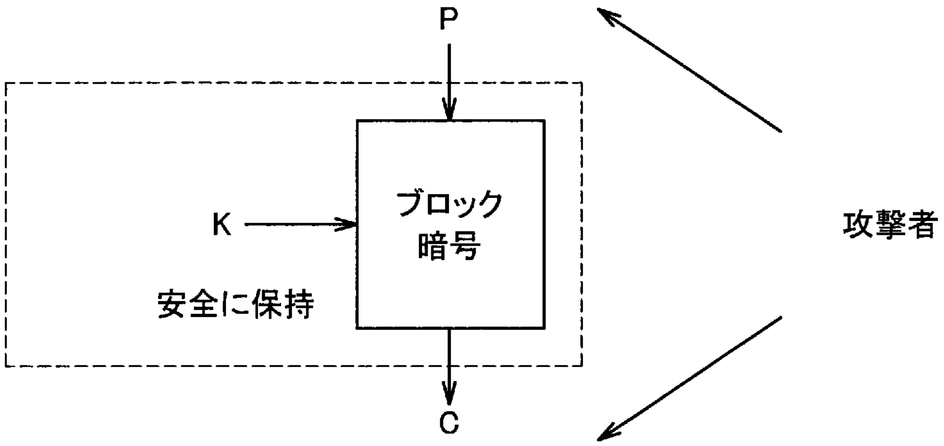
[図3]


第 i ラウンド
X_i の上位n/2ビット
X_i の下位n/2ビット
X_i
RK_i
F関数
X_{i+1} の上位n/2ビット
X_{i+1} の下位n/2ビット
X_{i+1}
X_1
X_i
X_{i+1}
X_{R+1}
ラウンド関数
ラウンド関数
ラウンド関数
ラウンド関数
RK_1
RK_2
RK_i
RK_iR

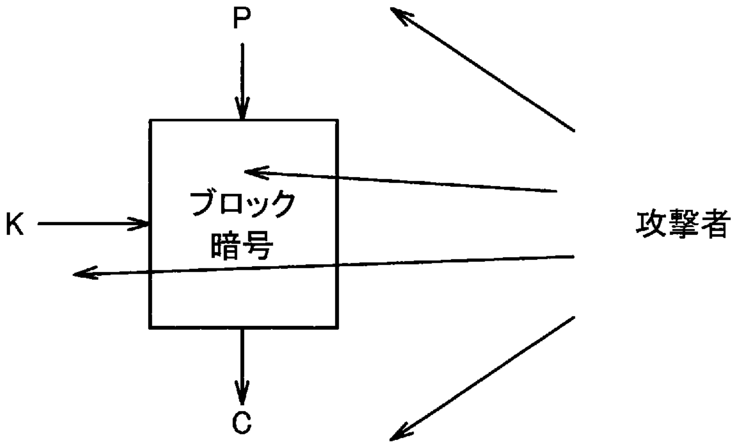
[図4]


X_i
RK_i
鍵排他的論理和部
X_iの第1 n/dビット
X_iの第2 n/dビット
X_iの第3 n/dビット
X_iの第d n/dビット
S関数
S関数
S関数
S関数
線形変換 L
X_{i+1}
X_1
RK_1
RK_2
RK_i
RK_{iR}
ラウンド関数
ラウンド関数
ラウンド関数
ラウンド関数
X_i
X_{i+1}
X_{R+1}

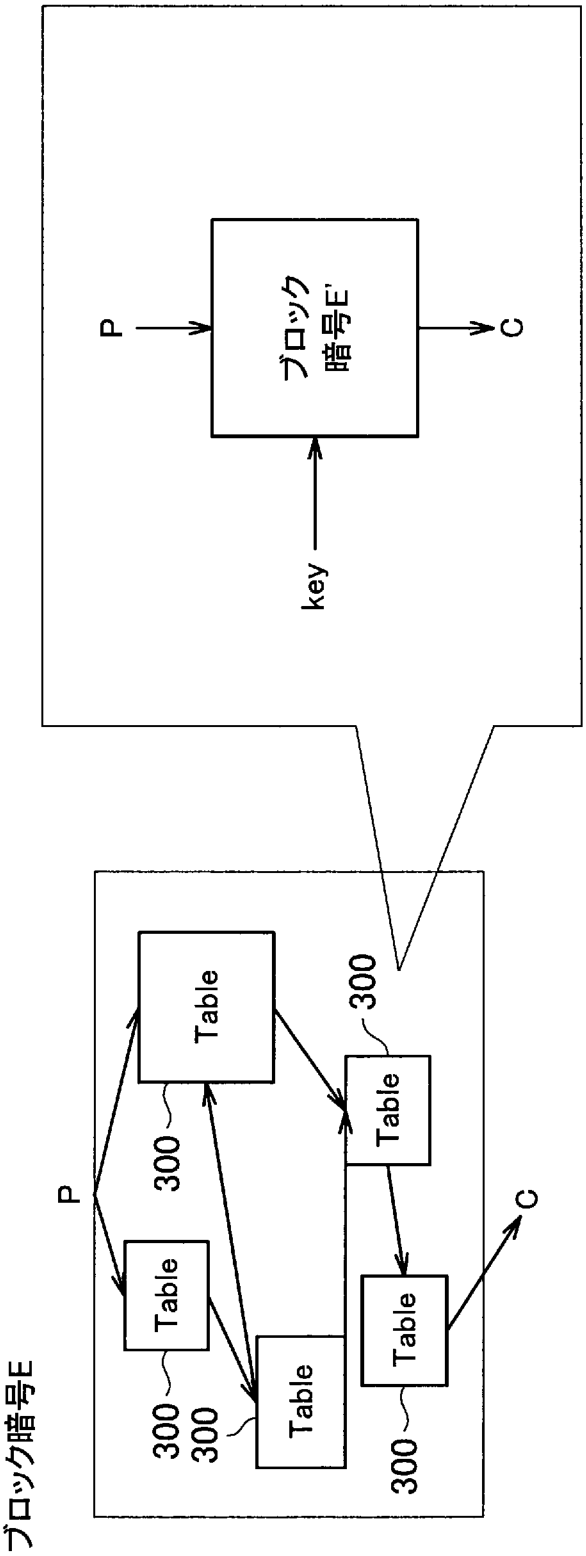
［図5］



［図6］



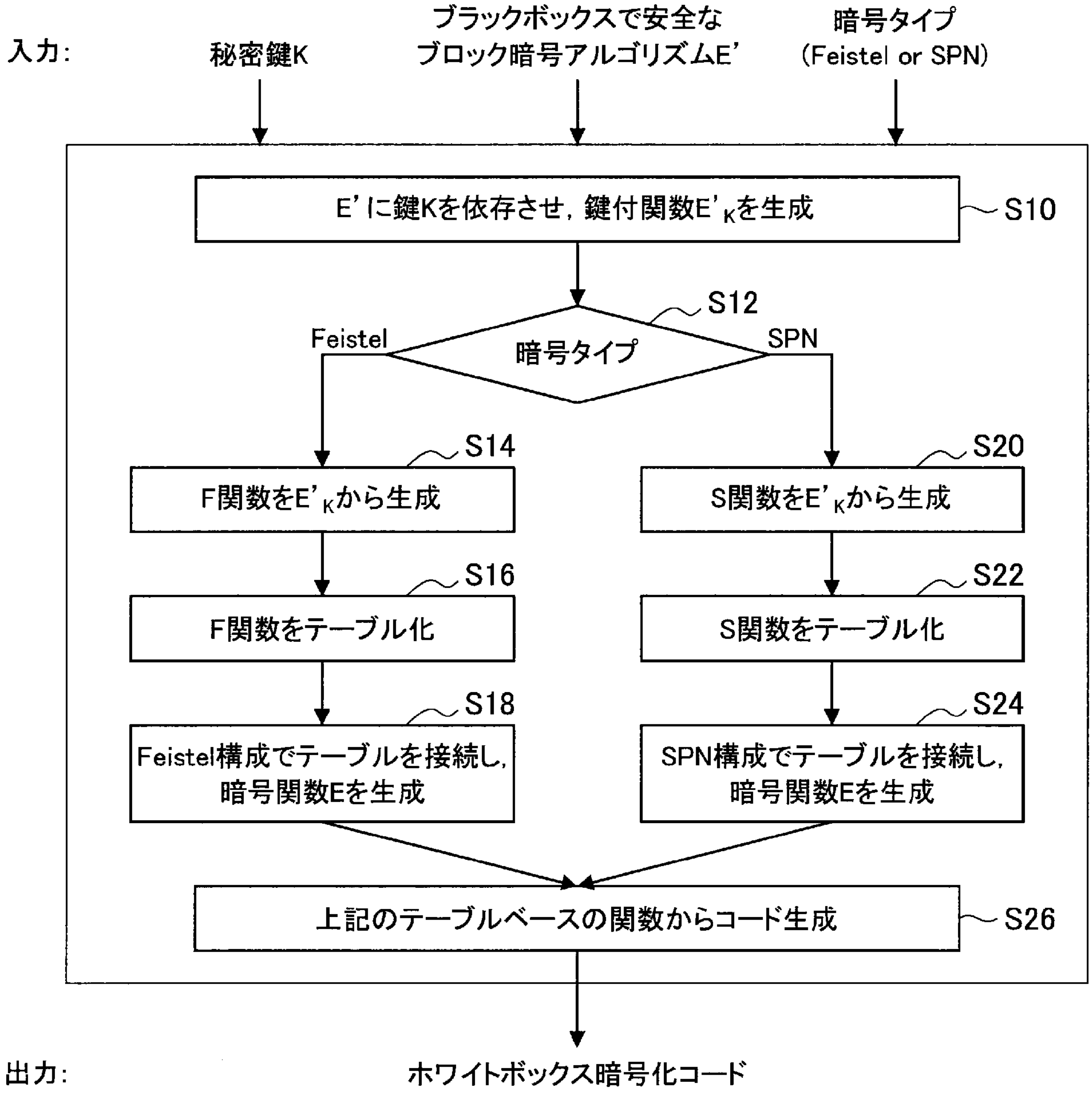
[図7]


P
ブロック
暗号E'
key
C
ブロック暗号E
P
Table
Table
Table
Table
Table
300
300
300
300
300
C

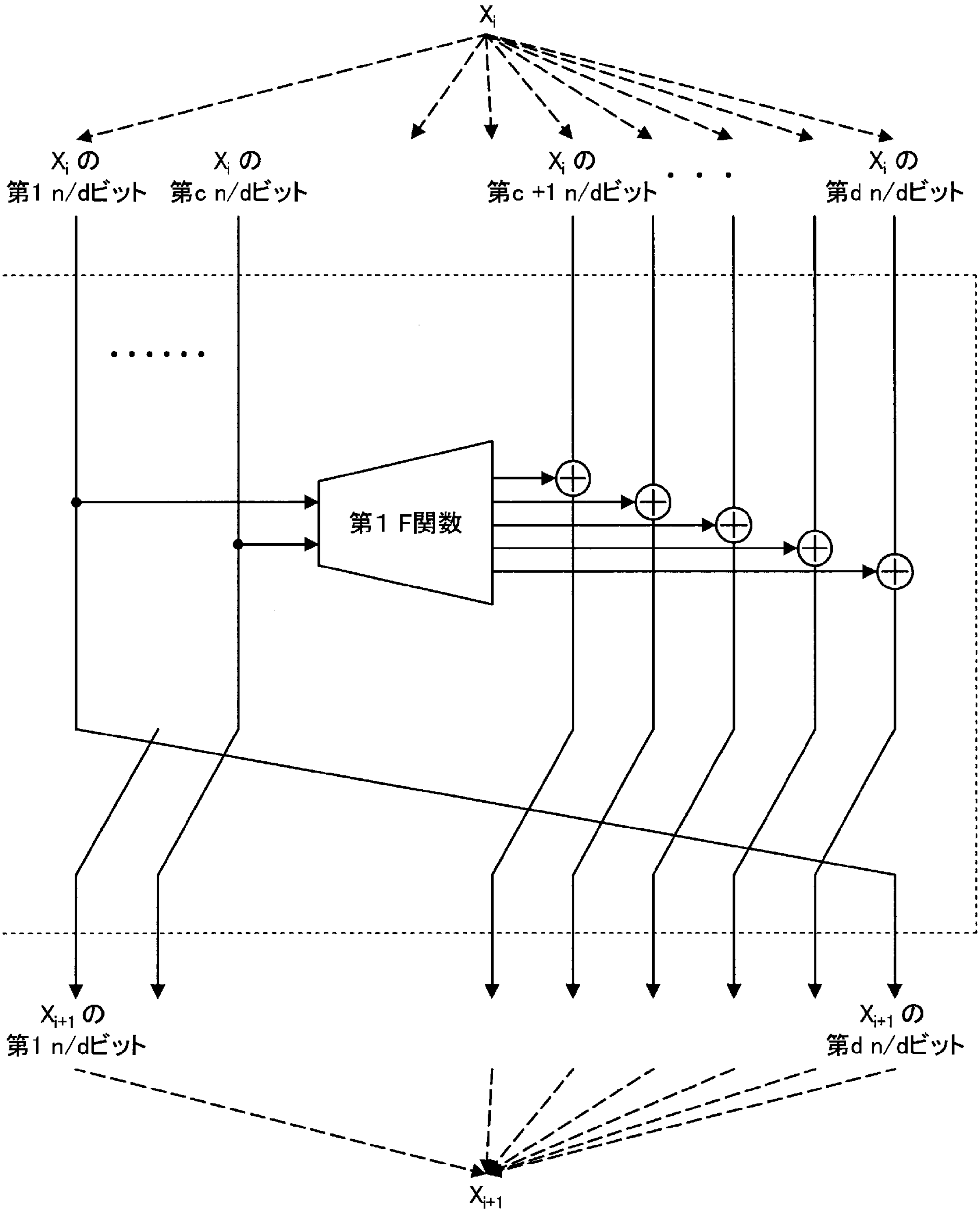
[図8]

	全体構成	F関数/S関数の種類	Table size可変
構成B	Feistel	1	No
構成C	SPN	1	No
構成D	Feistel	複数	Yes
構成E	SPN	複数	Yes

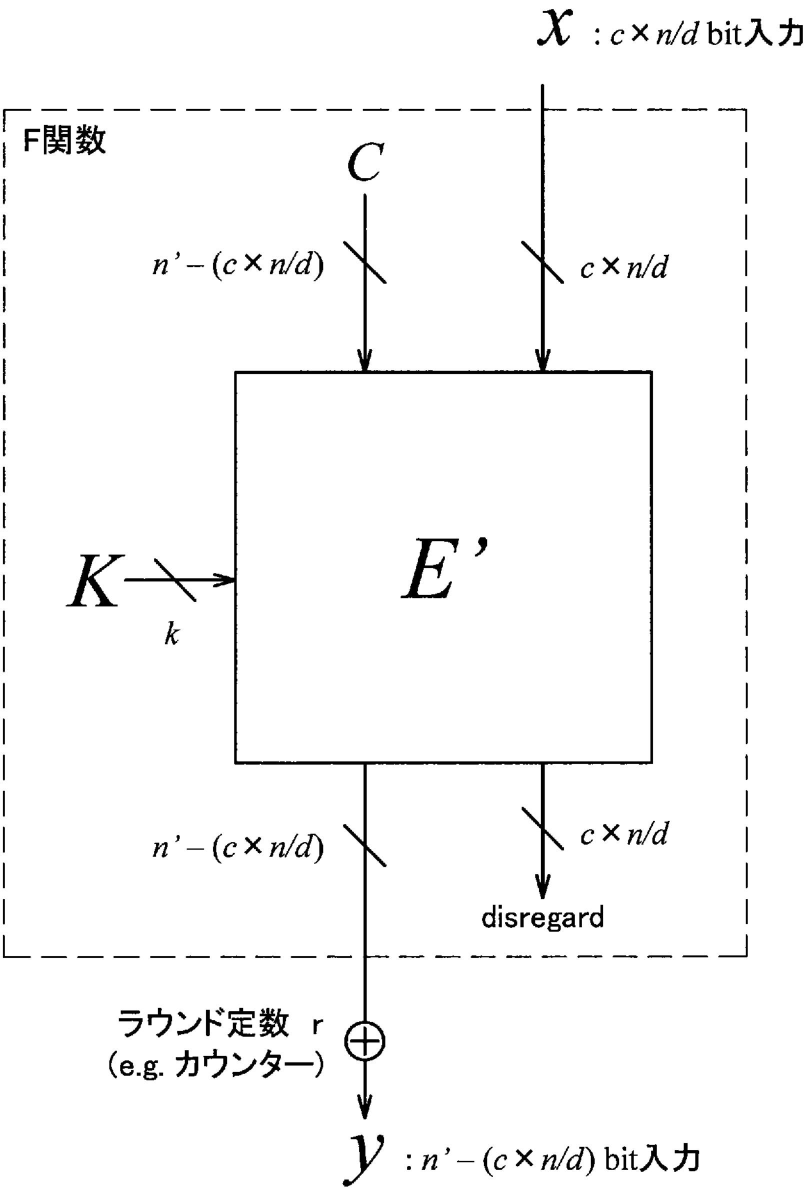
[図9]



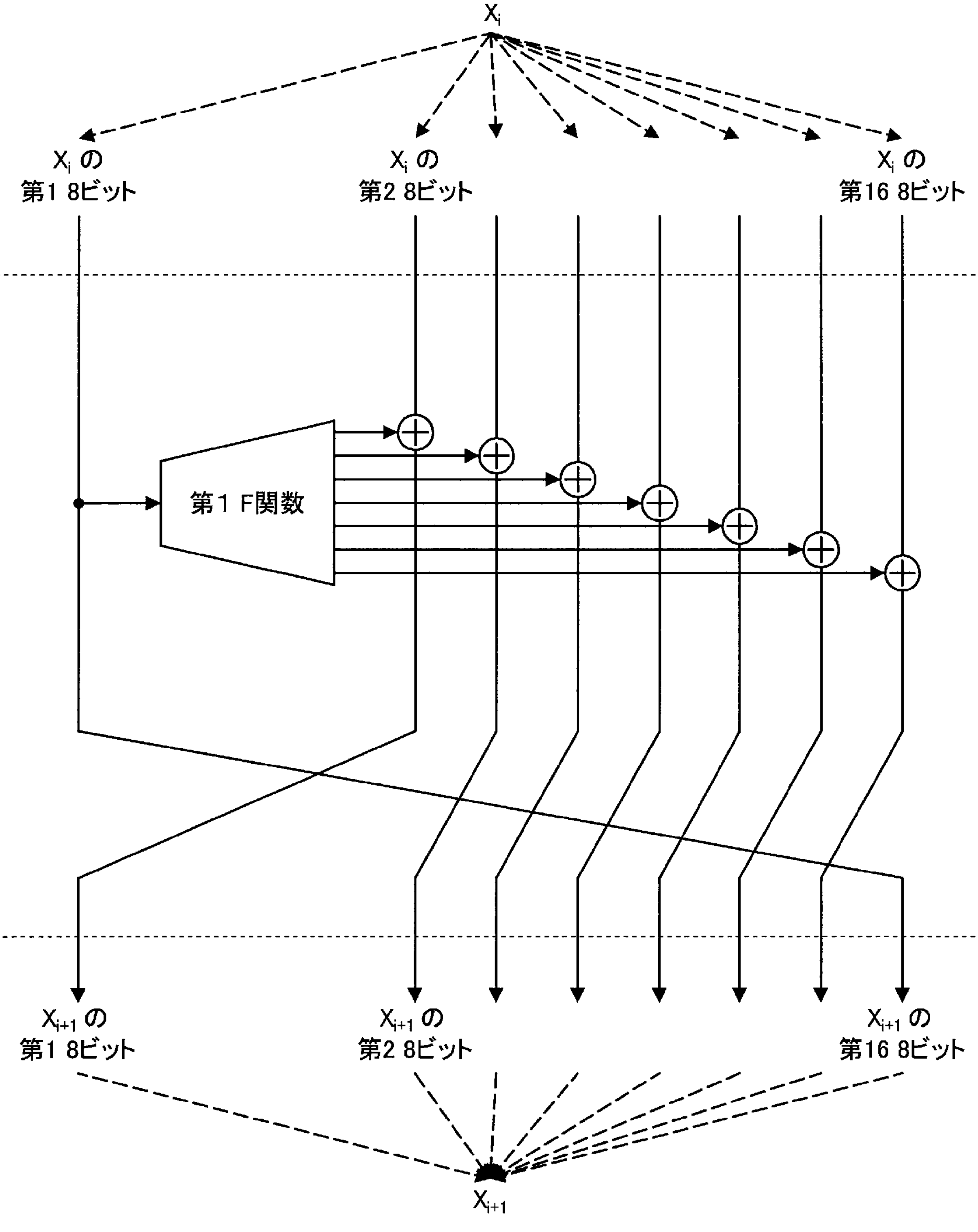
[図10]


X_i
X_i の
第1 n/dビット
X_i の
第c n/dビット
X_i の
第c +1 n/dビット
・・・
X_i の
第d n/dビット
・・・・・・
第1 F関数
X_{i+1} の
第1 n/dビット
X_{i+1} の
第d n/dビット
X_{i+1}

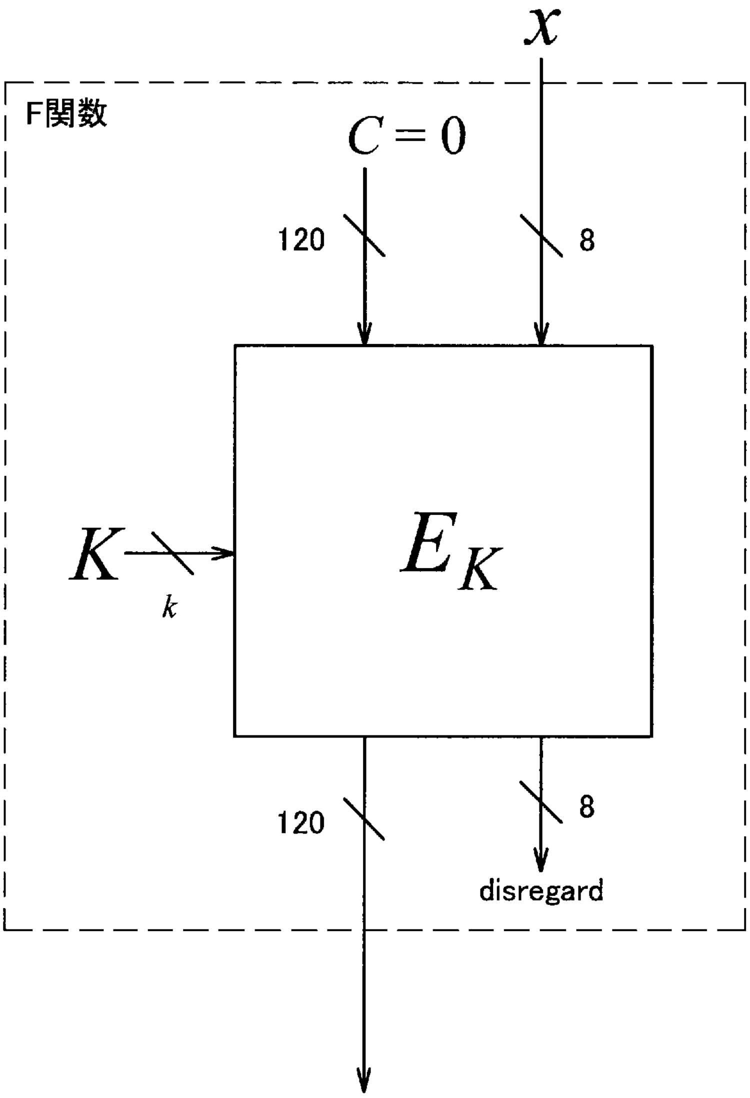
[図11]


x : $c \times n/d$ bit入力
F関数
C
$n' - (c \times n/d)$
$c \times n/d$
K
k
E'
$n' - (c \times n/d)$
$c \times n/d$
disregard
ラウンド定数 r
(e.g. カウンター)
y : $n' - (c \times n/d)$ bit入力

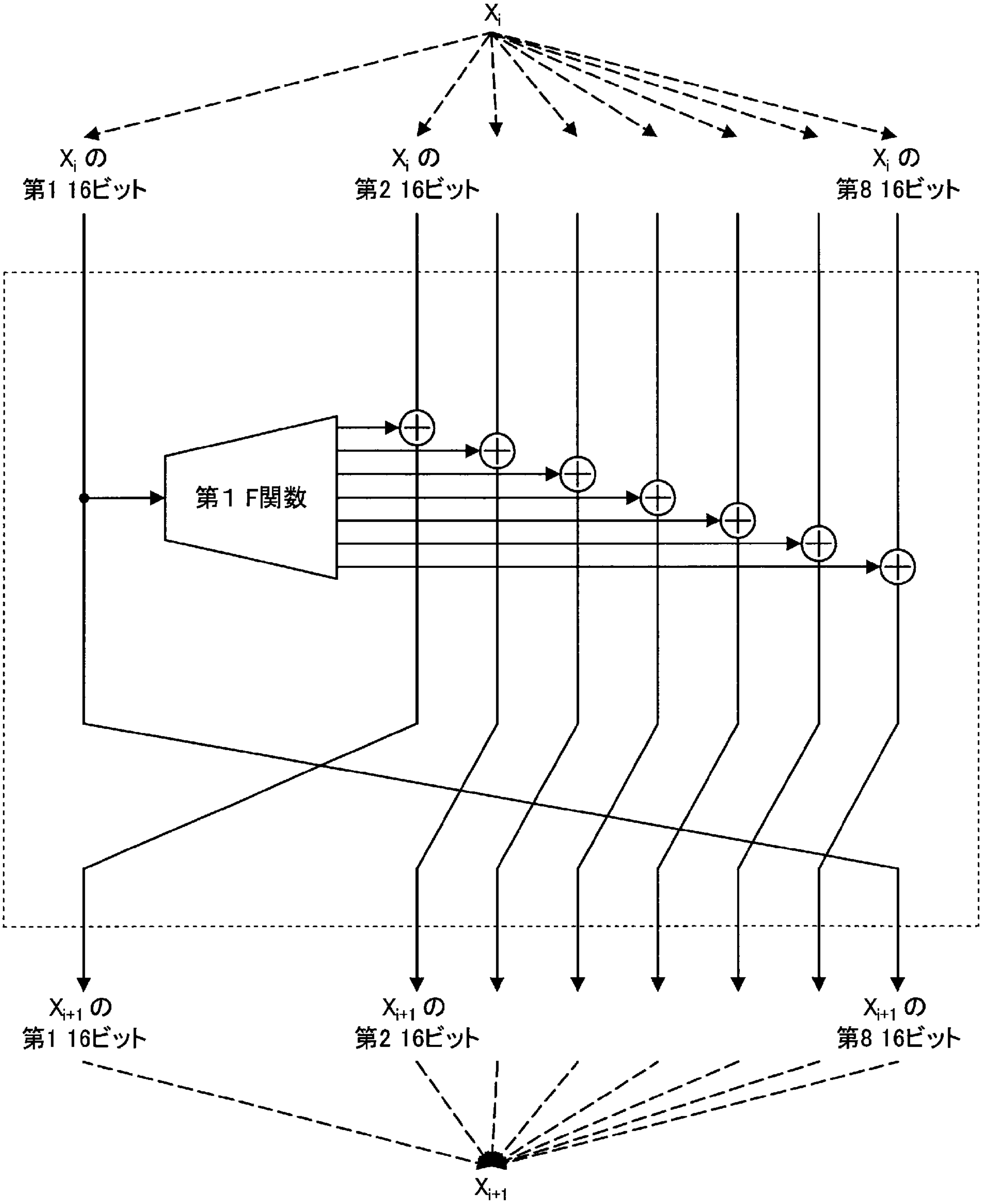
[図12]


X_i
X_i の
第1 8ビット
X_i の
第2 8ビット
X_i の
第16 8ビット
第１ F関数
X_{i+1} の
第1 8ビット
X_{i+1} の
第2 8ビット
X_{i+1} の
第16 8ビット
X_{i+1}

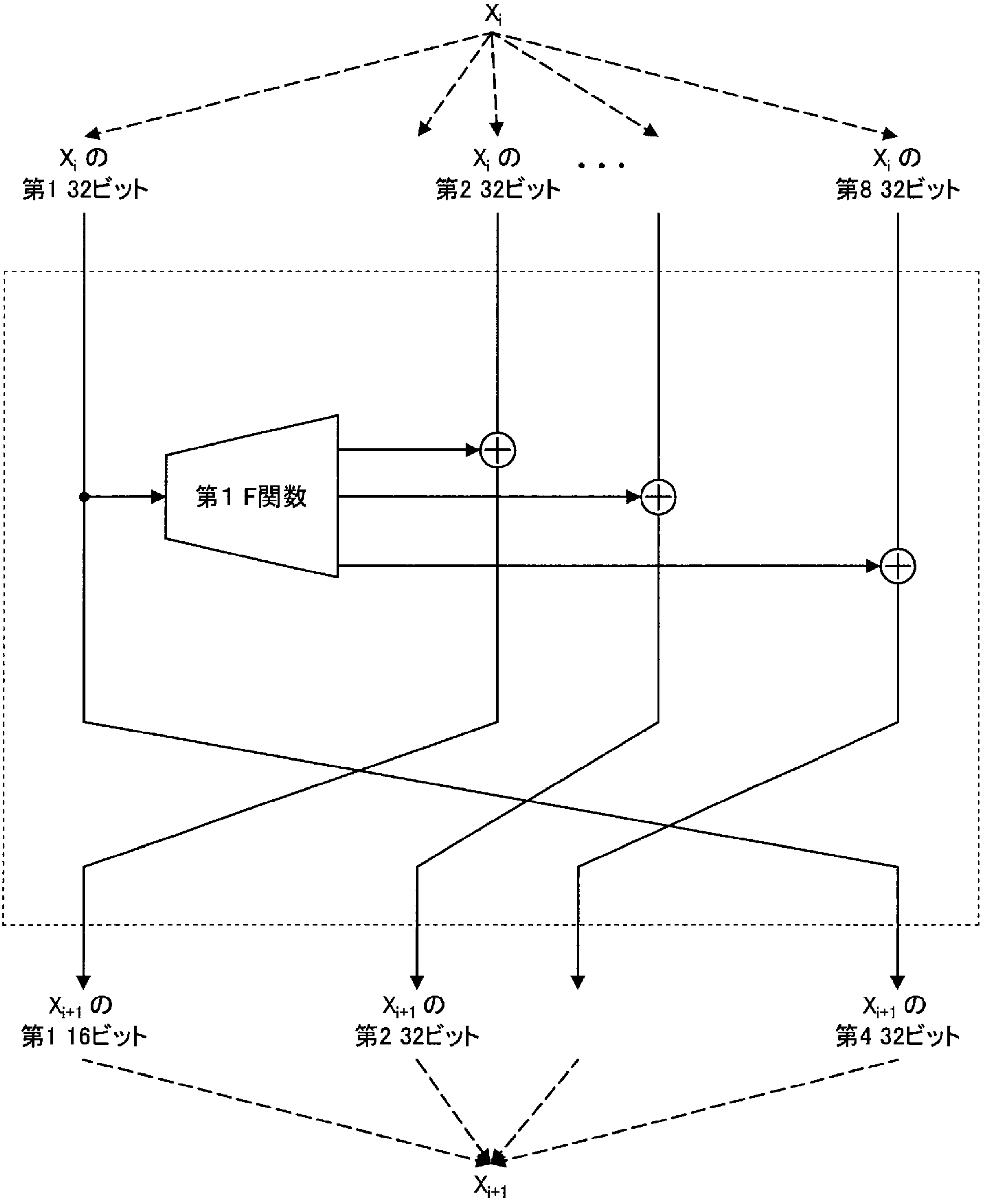
[図13]



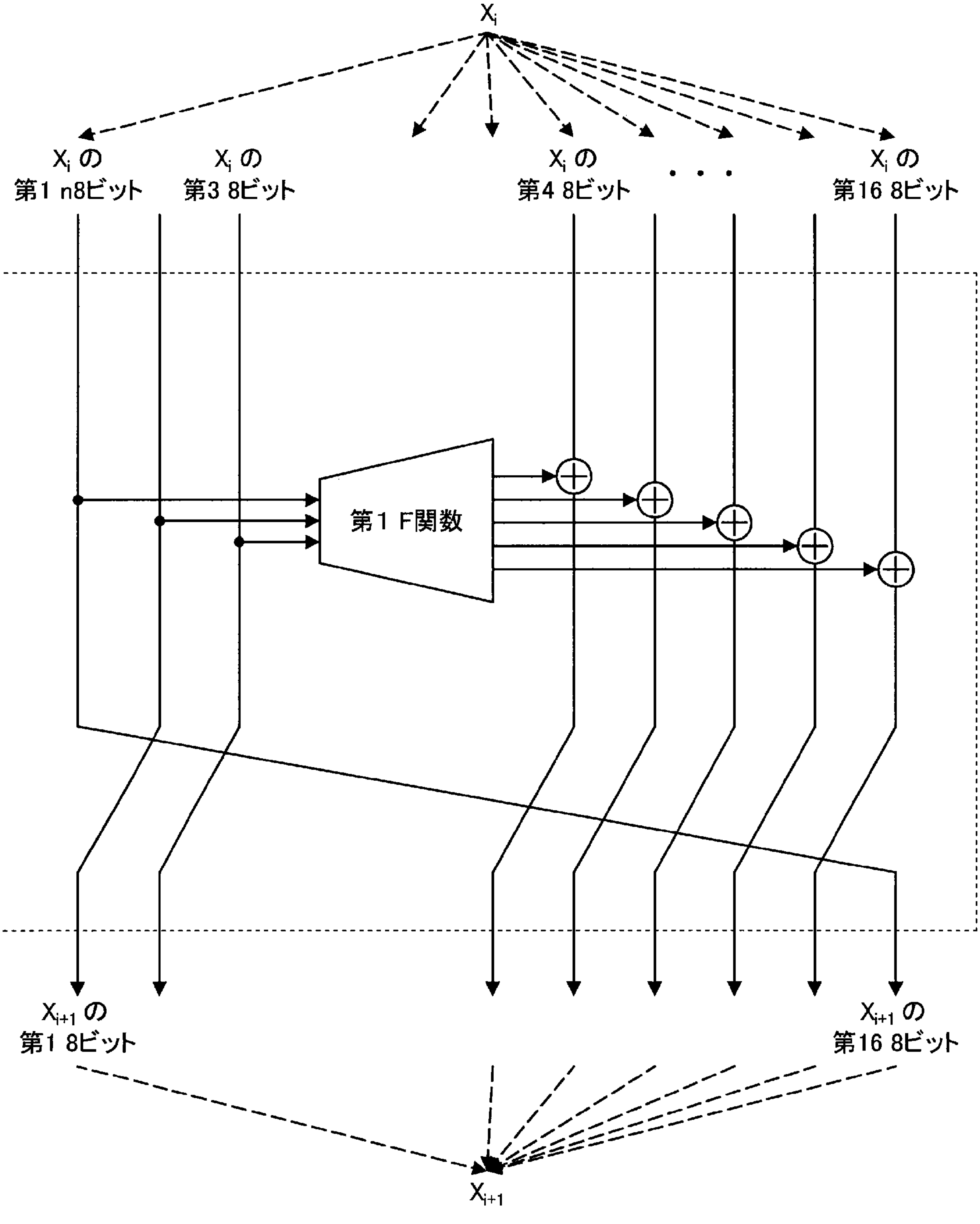
［図14］


X_i
X_i の
第1 16ビット
X_i の
第2 16ビット
X_i の
第8 16ビット
第１ F関数
X_{i+1} の
第1 16ビット
X_{i+1} の
第2 16ビット
X_{i+1} の
第8 16ビット
X_{i+1}

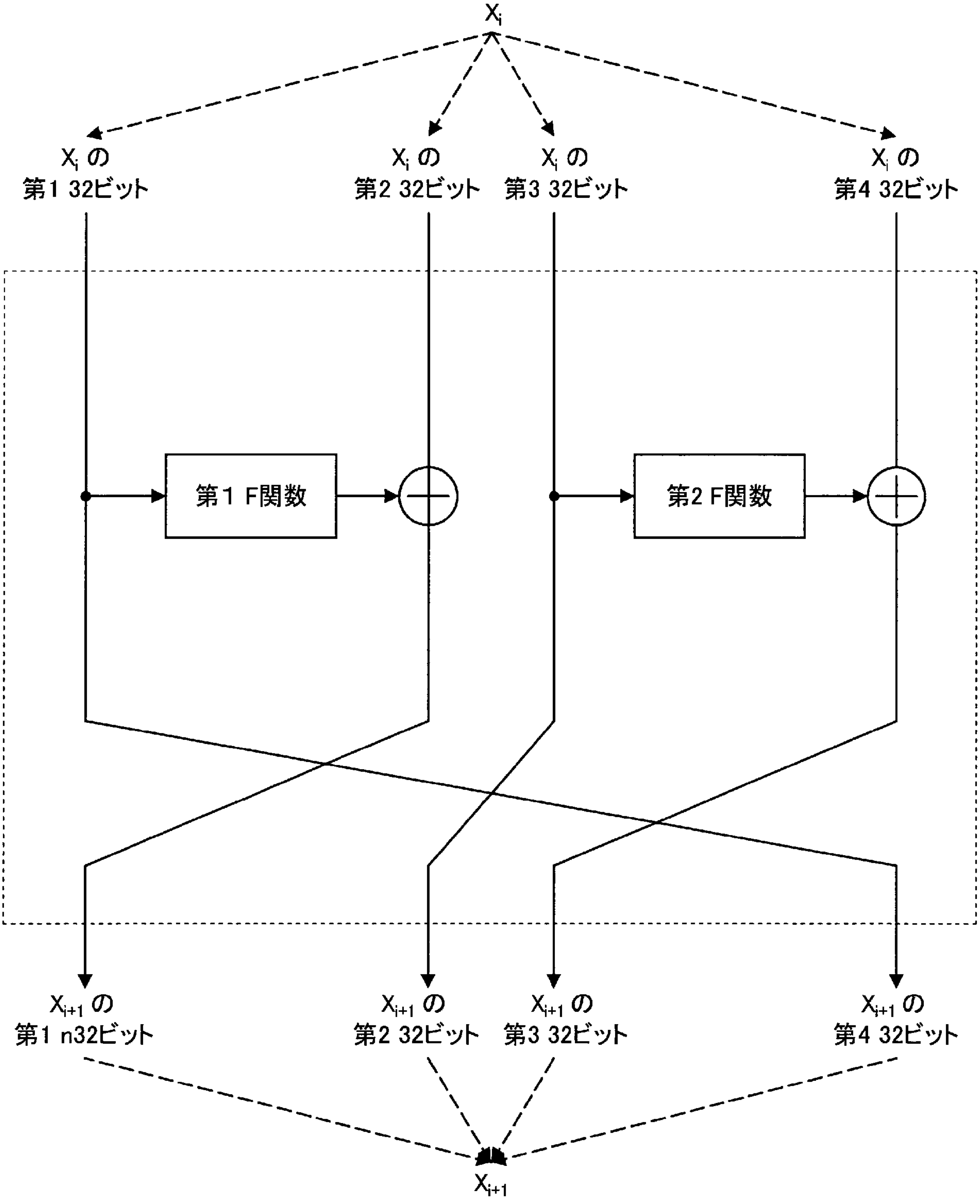
［図15］


X_i
X_i の
第1 32ビット
X_i の
第2 32ビット
・・・
X_i の
第8 32ビット
第１ F関数
X_{i+1} の
第1 16ビット
X_{i+1} の
第2 32ビット
X_{i+1} の
第4 32ビット
X_{i+1}

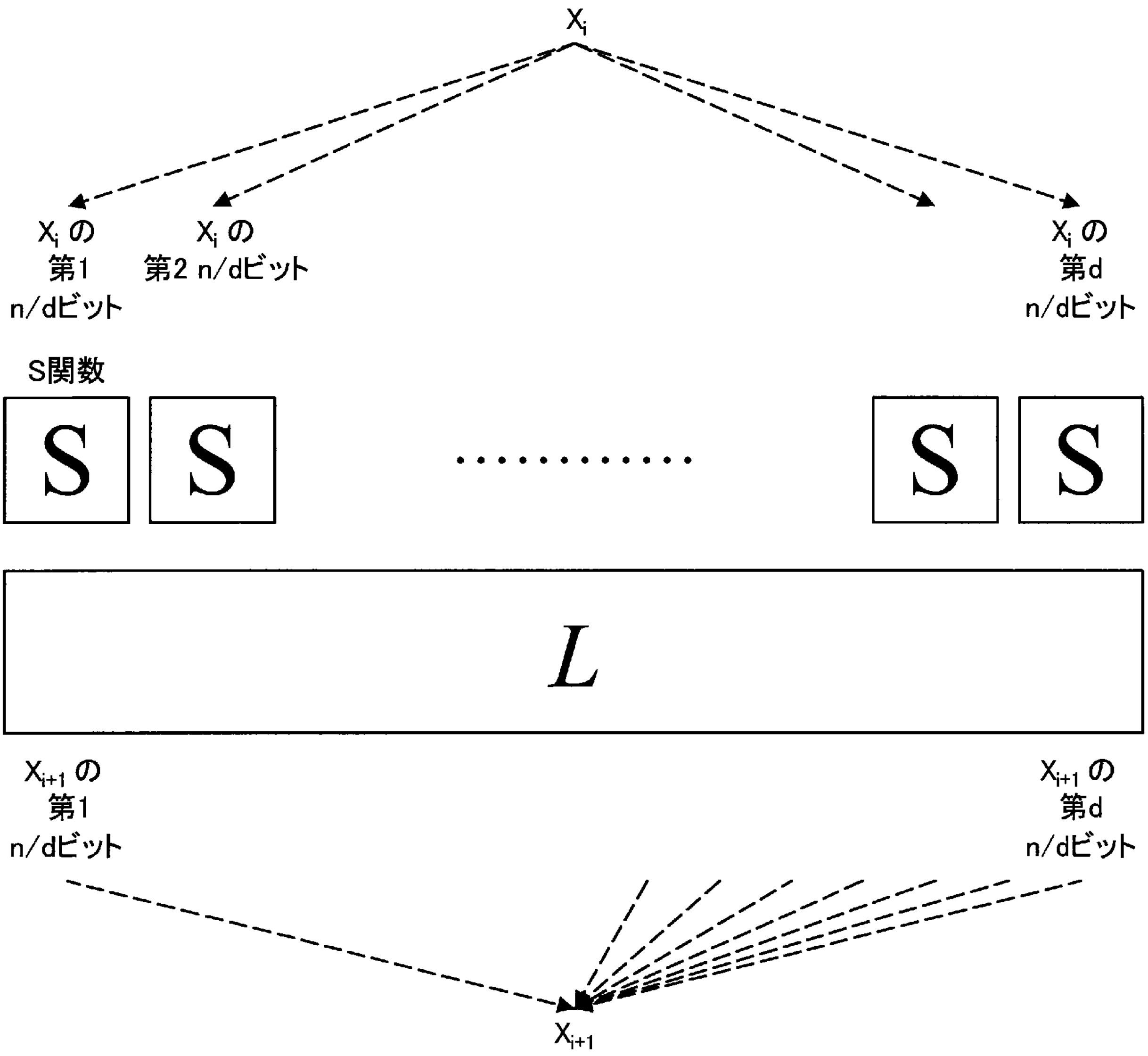
［図16］


X_i
X_i の
第1 n8ビット
X_i の
第3 8ビット
X_i の
第4 8ビット
・・・
X_i の
第16 8ビット
第1 F関数
X_{i+1} の
第1 8ビット
X_{i+1} の
第16 8ビット
X_{i+1}

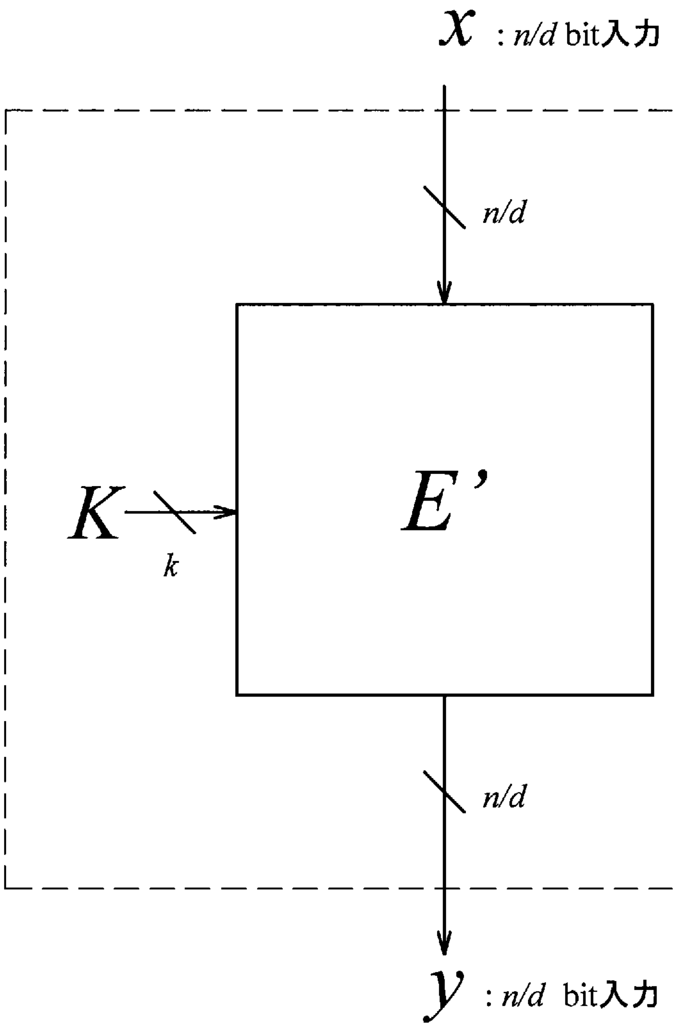
［図17］


X_i
X_i の
第1 32ビット
X_i の
第2 32ビット
X_i の
第3 32ビット
X_i の
第4 32ビット
第１ F関数
第2 F関数
X_{i+1} の
第1 n32ビット
X_{i+1} の
第2 32ビット
X_{i+1} の
第3 32ビット
X_{i+1} の
第4 32ビット
X_{i+1}

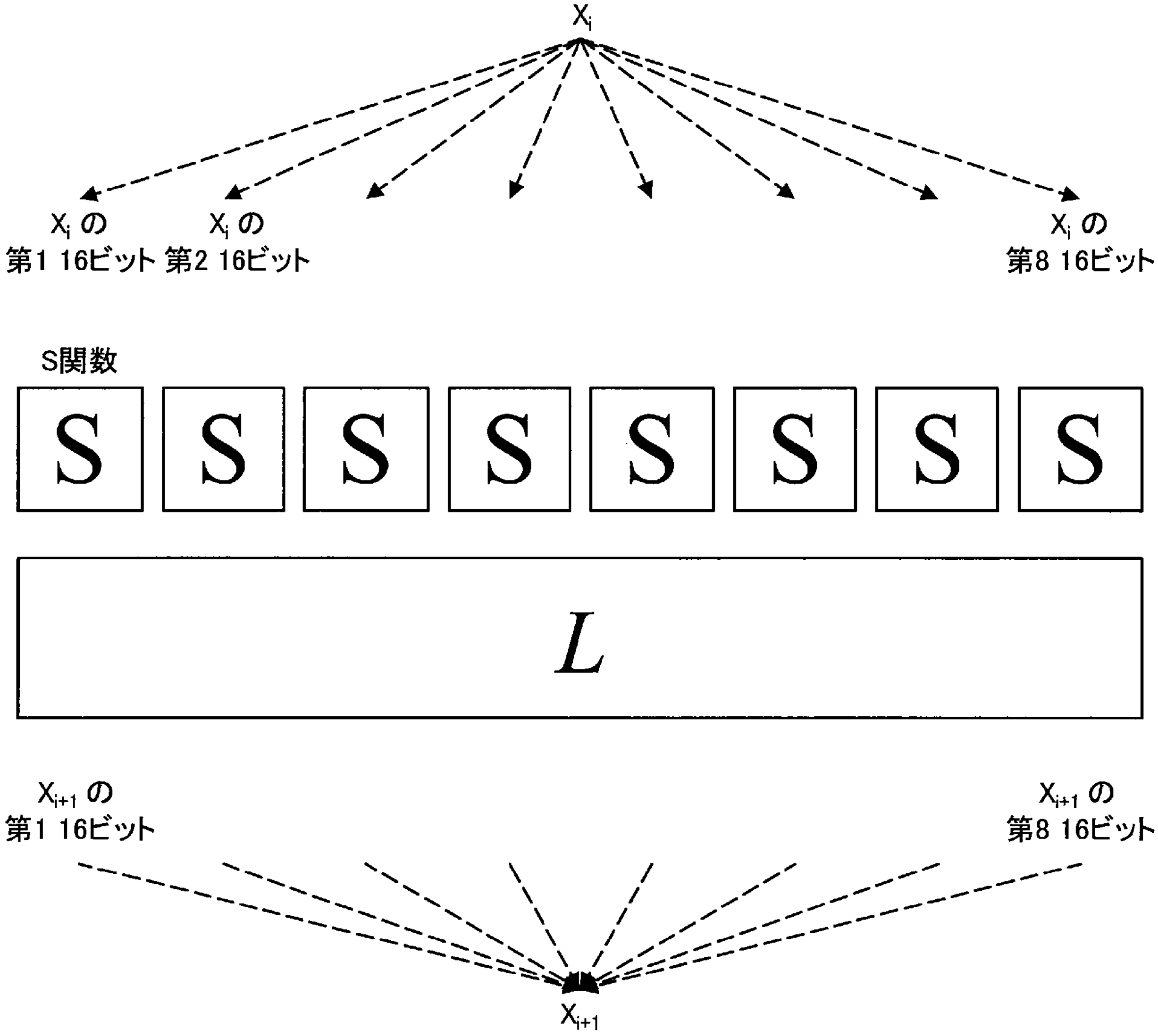
[図18]


X_i
X_i の 第1 n/dビット
X_i の 第2 n/dビット
X_i の 第d n/dビット
S関数
S
S
S
S
L
X_{i+1} の 第1 n/dビット
X_{i+1} の 第d n/dビット
X_{i+1}

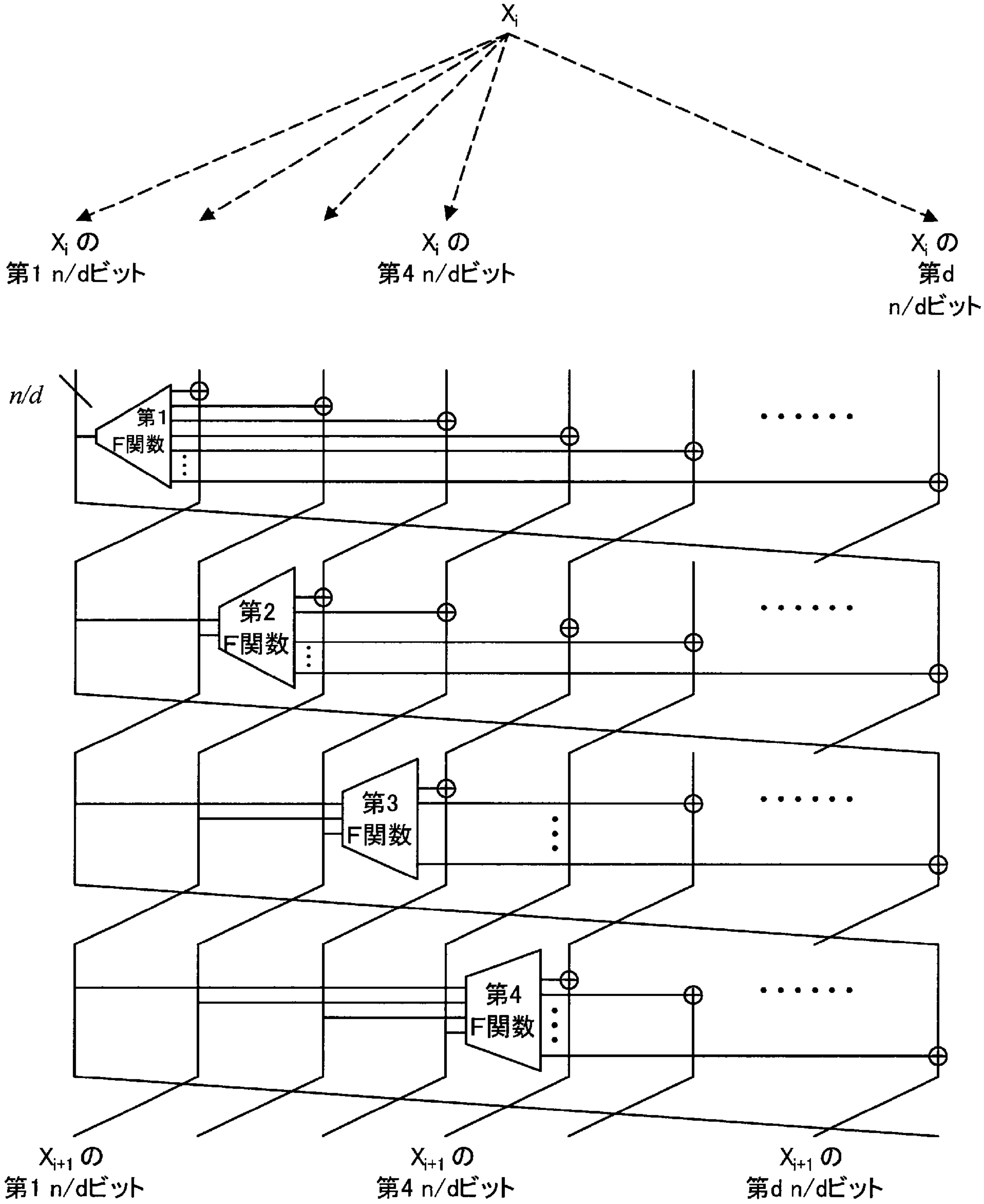
[図19]



[図20]

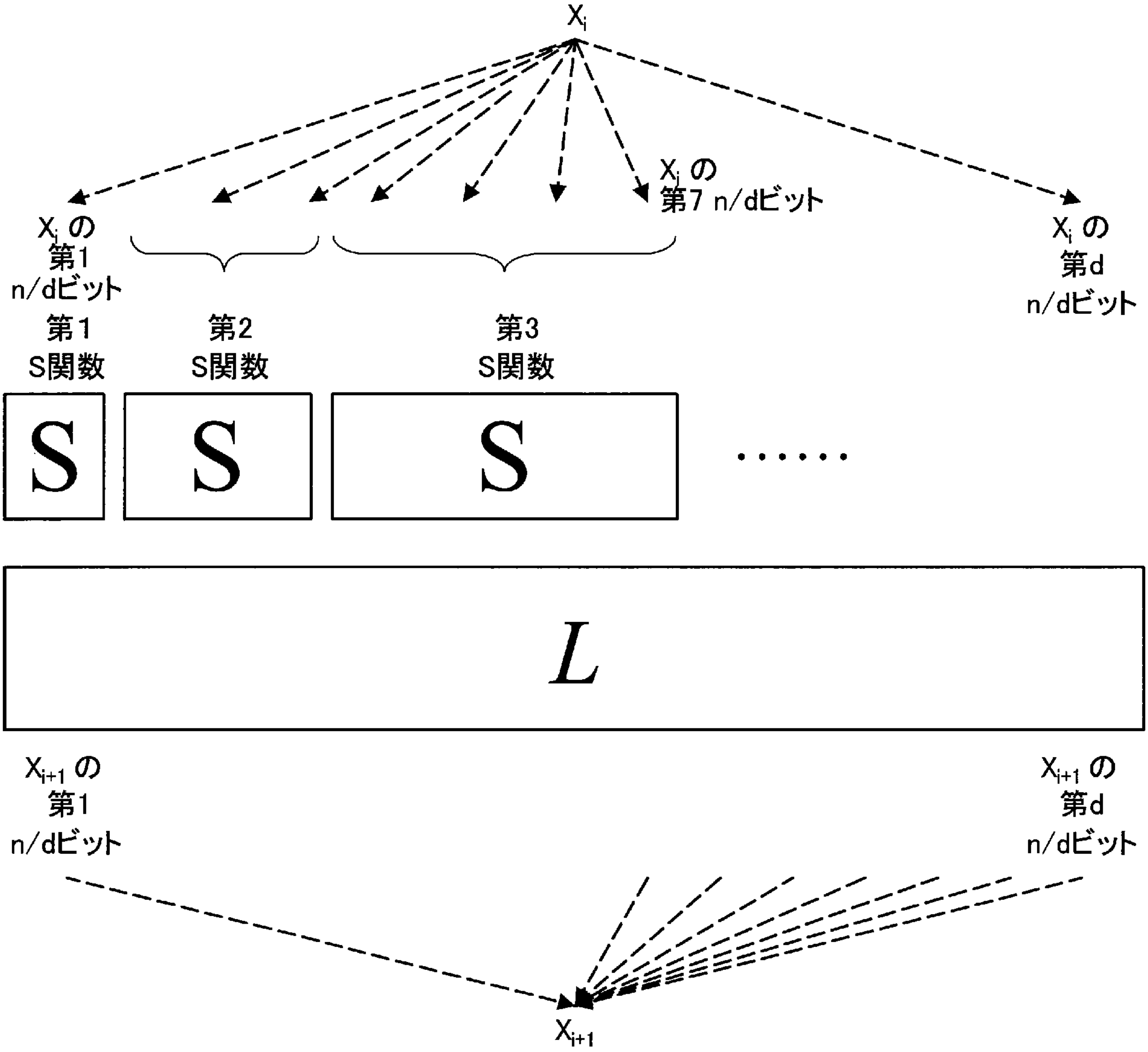

X_i
X_i の 第1 16ビット
X_i の 第2 16ビット
X_i の 第8 16ビット
S関数
S
S
S
S
S
S
S
S
L
X_{i+1} の 第1 16ビット
X_{i+1} の 第8 16ビット
X_{i+1}

[図21]



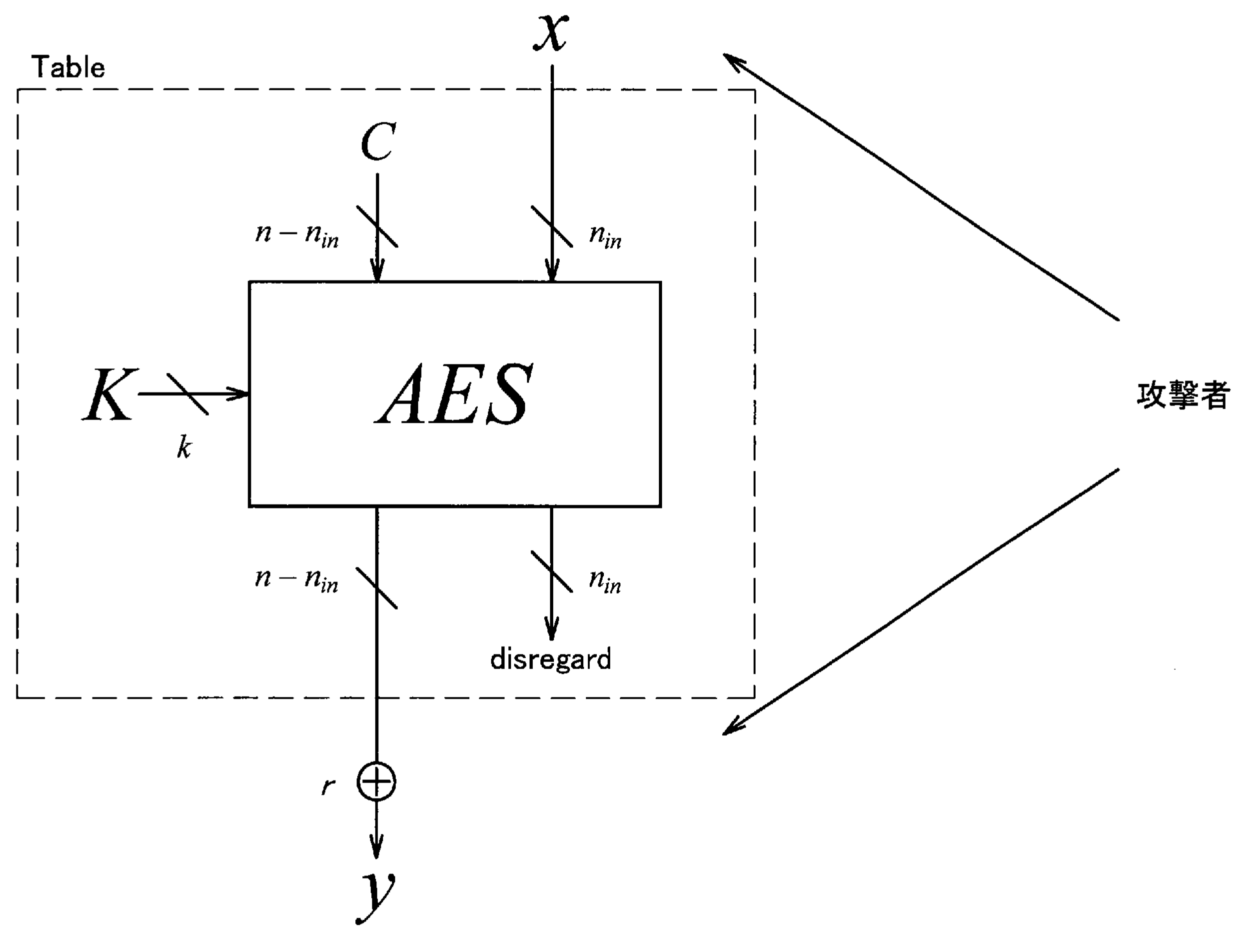
第1 F関数 (n/d bit 入力, n − n/d bit 出力)
第2 F関数 (2n/d bit 入力, n − 2n/d bit 出力)
第3 F関数 (3n/d bit 入力, n − 3n/d bit 出力)
第4 F関数 (4n/d bit入力, n − 4n/d bit 出力

[図22]

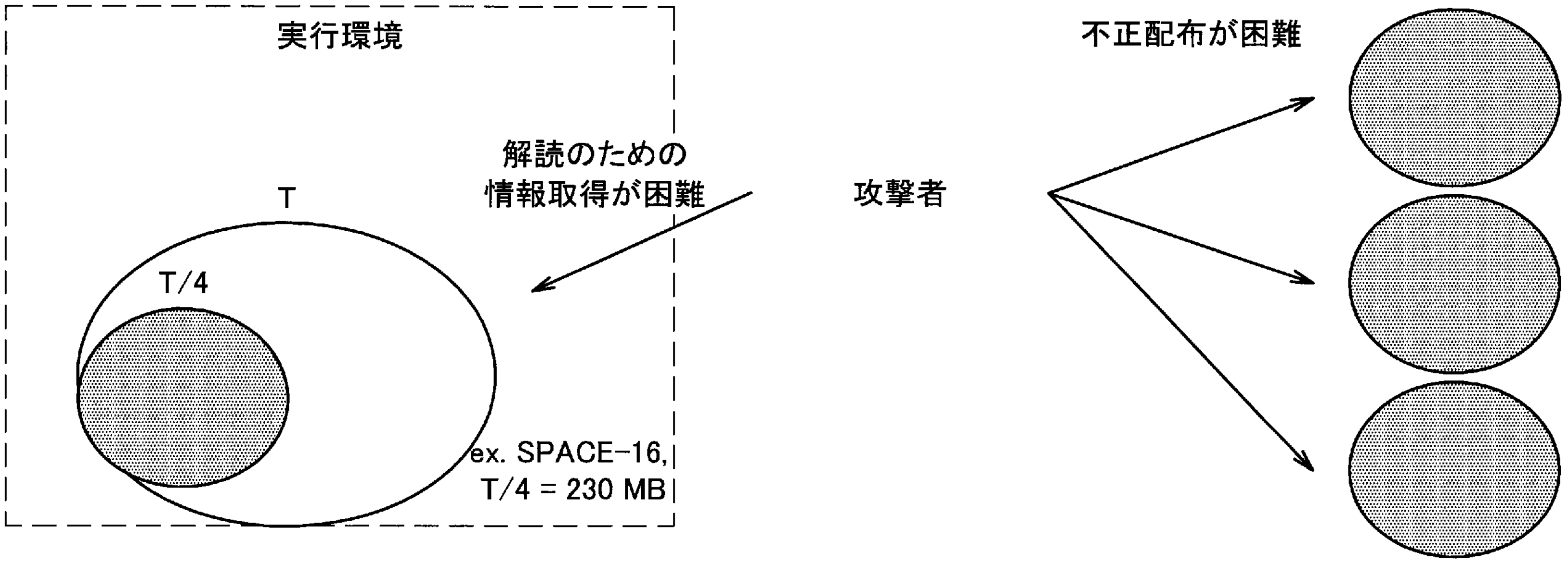


第1 S関数 (n/d bit to n − n/d bit)
第2 S関数 (2n/d bit to n − 2n/d bit)
第3 S関数 (4n/d bit to n − 4n/d bit)

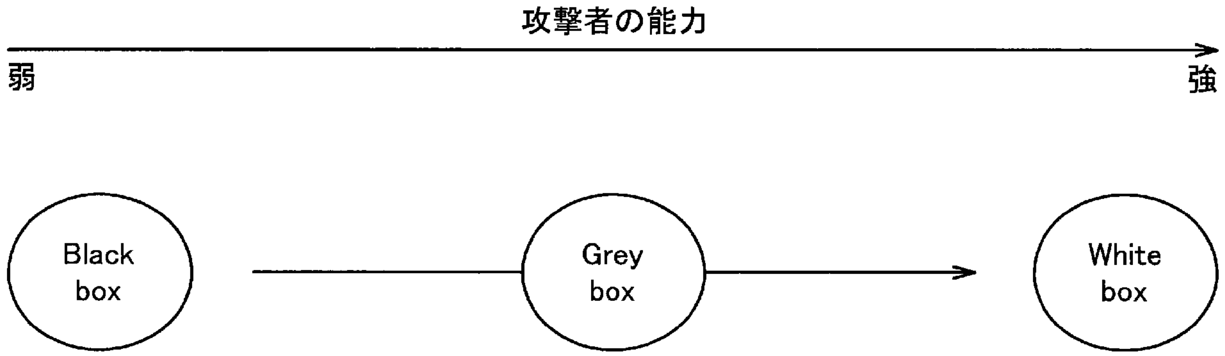
[図23]



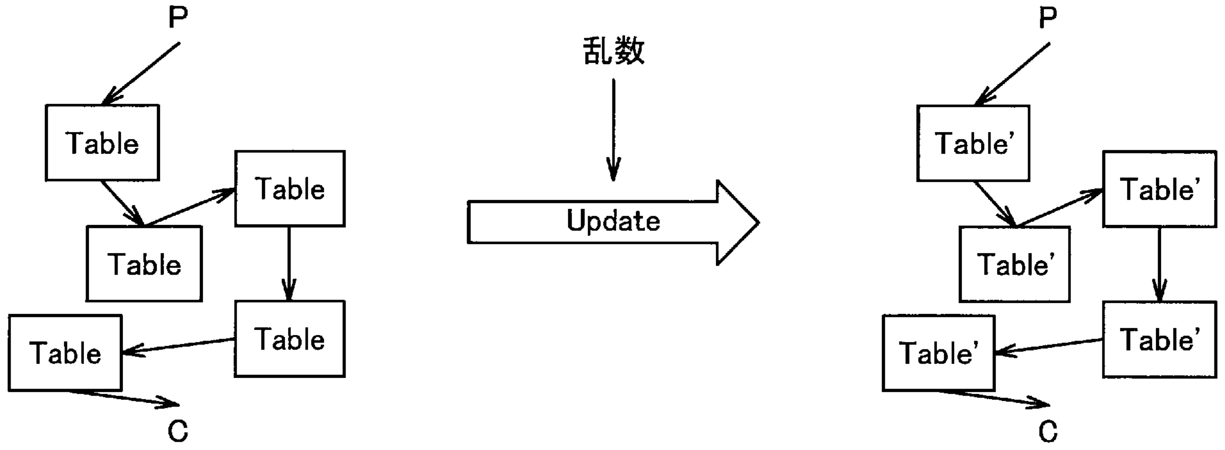
[図24]



[図25]

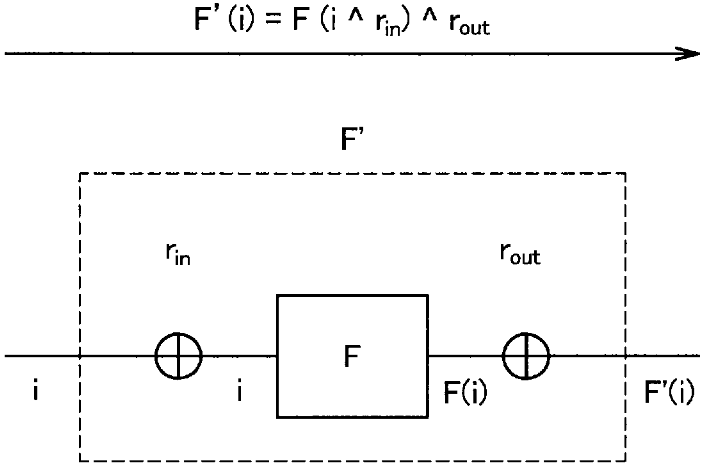

攻撃者の能力
弱
強
Black box
Grey box
White box

[図26]


P
Table
Table
Table
Table
Table
C
乱数
Update
P
Table'
Table'
Table'
Table'
Table'
C

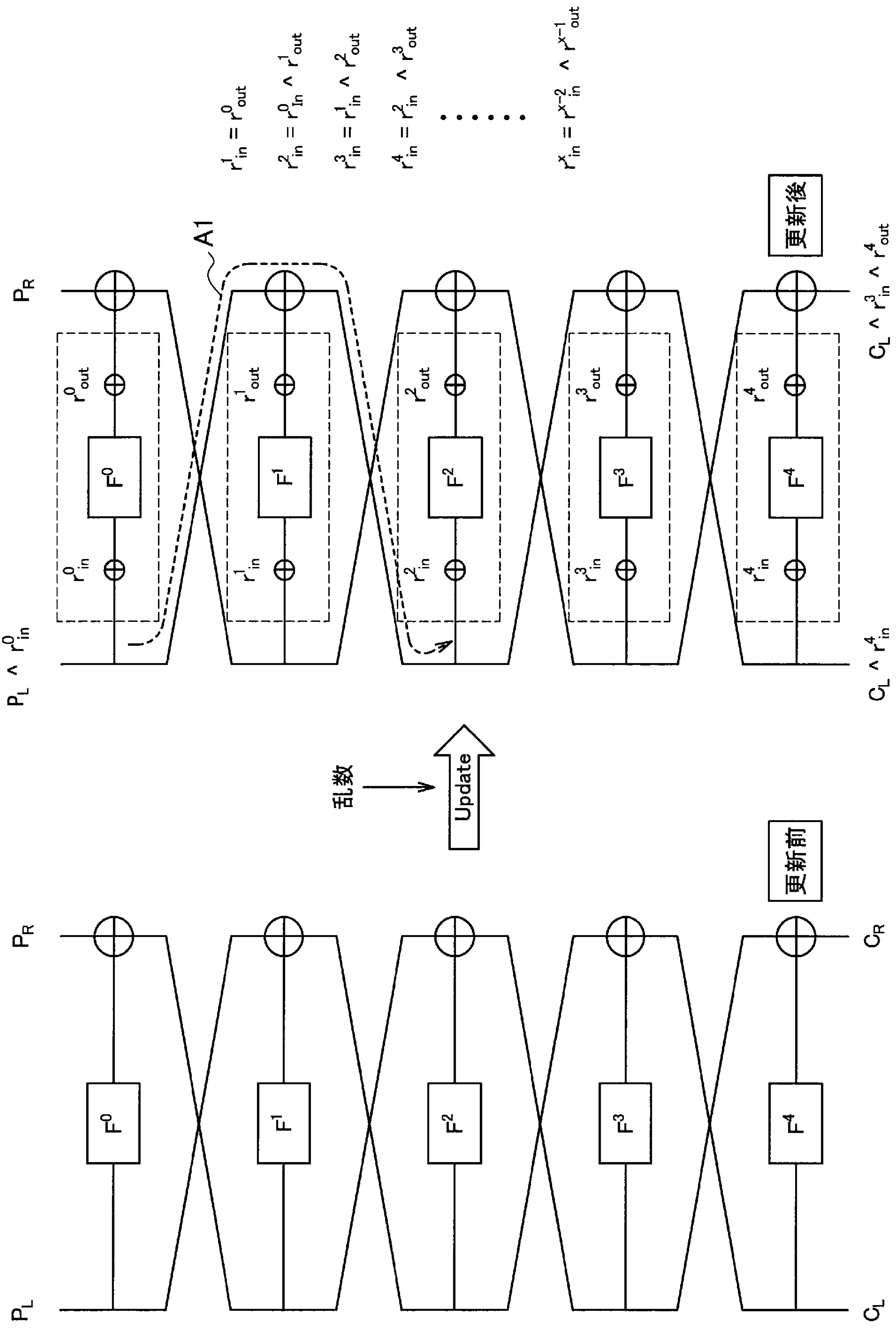
[図27]

i	F(i)
0	X
1	Y
2	Z
⋮	⋮
2^n-1	W

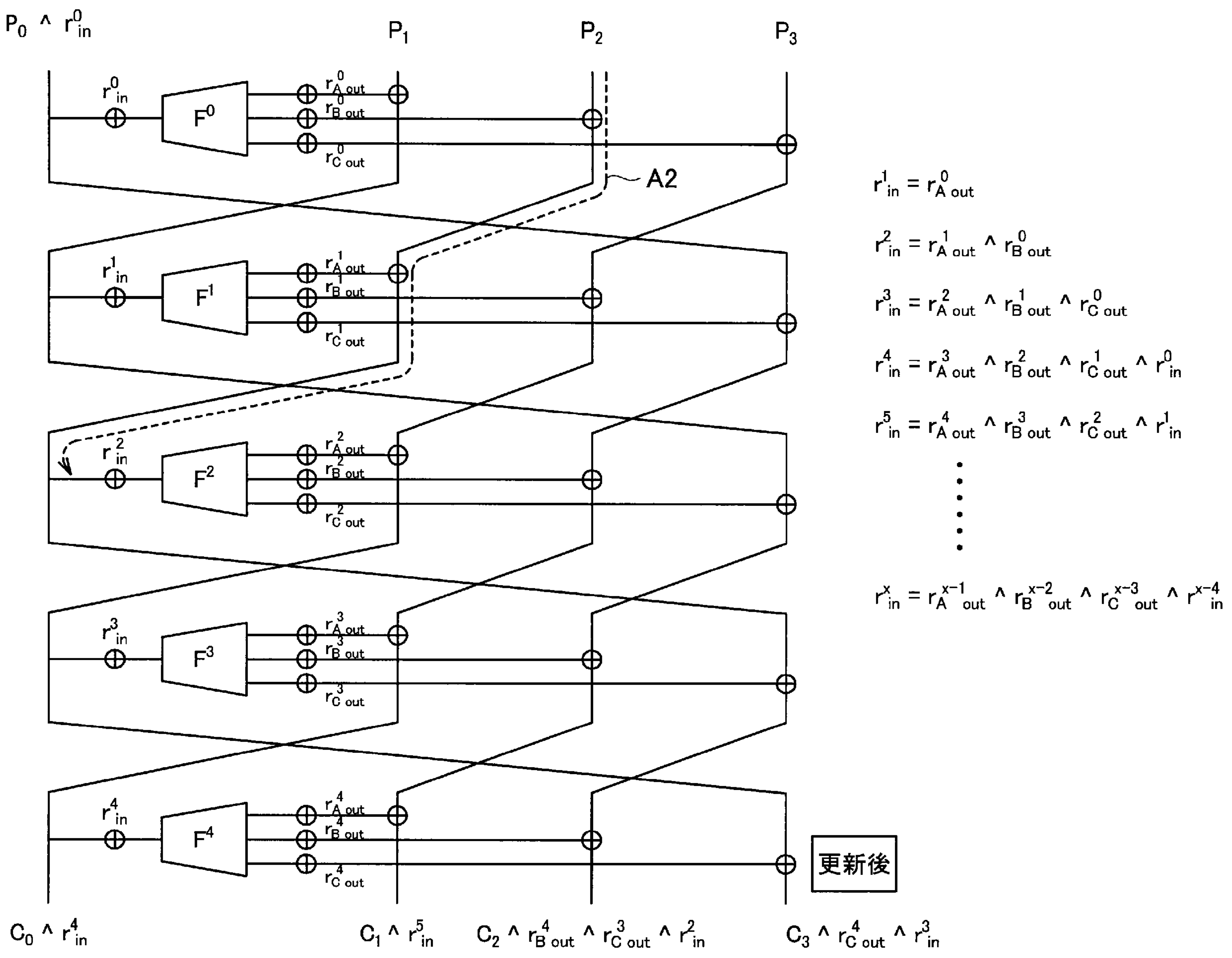


i	F'(i)
0	W
1	Y
2	X
⋮	⋮
2^n-1	Z

[図28]


更新前
P_L
P_R
F^0
F^1
F^2
F^3
F^4
C_L
C_R
乱数
Update
更新後
$P_L \wedge r^0_{in}$
r^0_{in}
r^0_{out}
r^1_{in}
r^1_{out}
r^2_{in}
r^2_{out}
r^3_{in}
r^3_{out}
r^4_{in}
r^4_{out}
A1
$C_L \wedge r^4_{in}$
$C_L \wedge r^3_{in} \wedge r^4_{out}$
$r^1_{in} = r^0_{out}$
$r^2_{in} = r^0_{in} \wedge r^1_{out}$
$r^3_{in} = r^1_{in} \wedge r^2_{out}$
$r^4_{in} = r^2_{in} \wedge r^3_{out}$
・・・・・・
$r^x_{in} = r^{x-2}_{in} \wedge r^{x-1}_{out}$

[図29]



$$r_{in}^1 = r_{A\,out}^0$$

$$r_{in}^2 = r_{A\,out}^1 \wedge r_{B\,out}^0$$

$$r_{in}^3 = r_{A\,out}^2 \wedge r_{B\,out}^1 \wedge r_{C\,out}^0$$

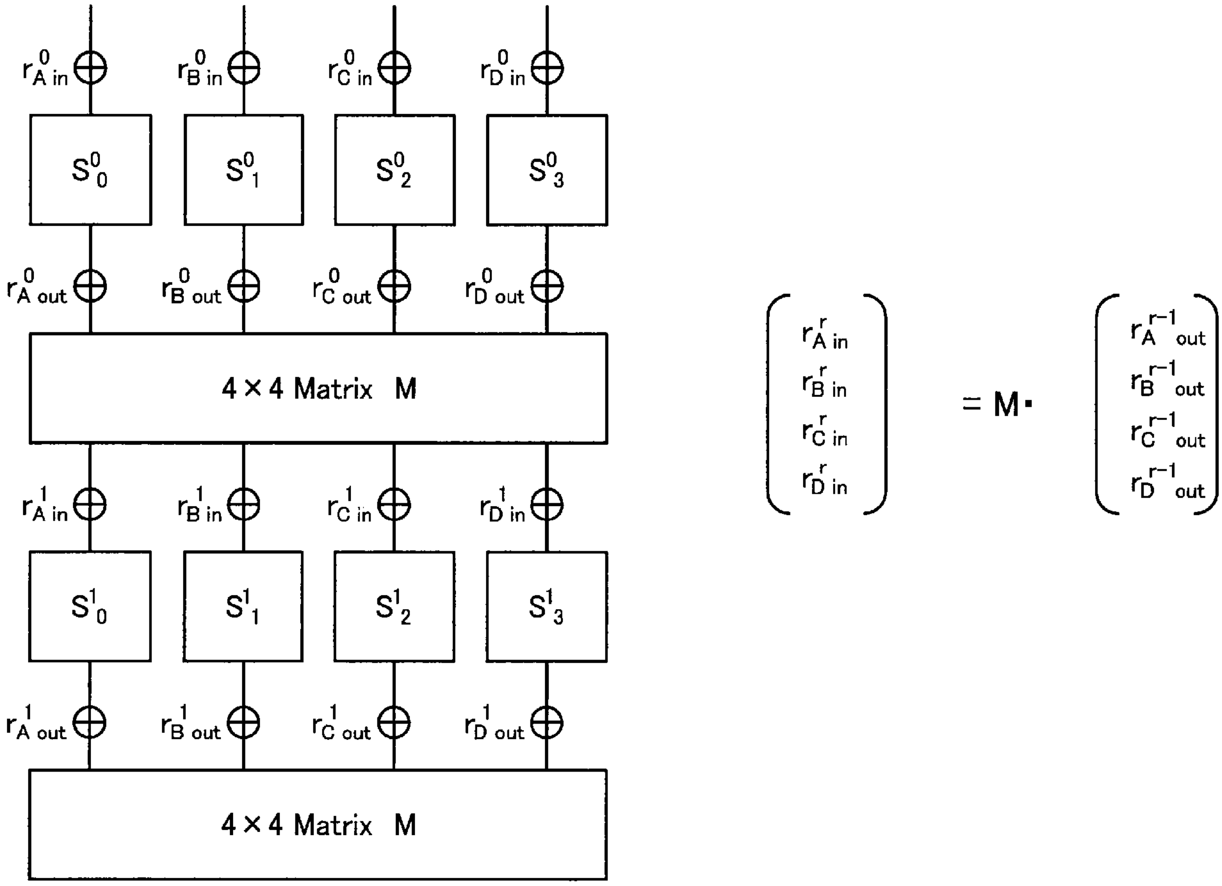
$$r_{in}^4 = r_{A\,out}^3 \wedge r_{B\,out}^2 \wedge r_{C\,out}^1 \wedge r_{in}^0$$

$$r_{in}^5 = r_{A\,out}^4 \wedge r_{B\,out}^3 \wedge r_{C\,out}^2 \wedge r_{in}^1$$

$$\vdots$$

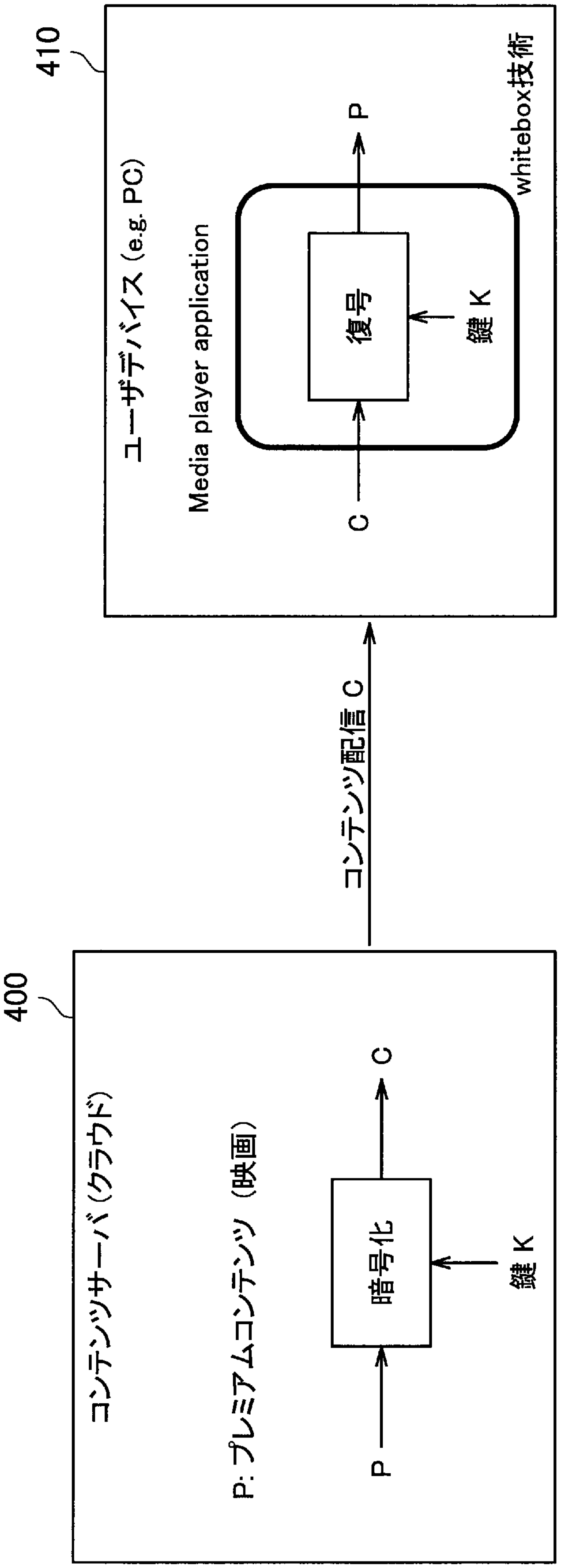
$$r_{in}^x = r_{A\,out}^{x-1} \wedge r_{B\,out}^{x-2} \wedge r_{C\,out}^{x-3} \wedge r_{in}^{x-4}$$

[図30]

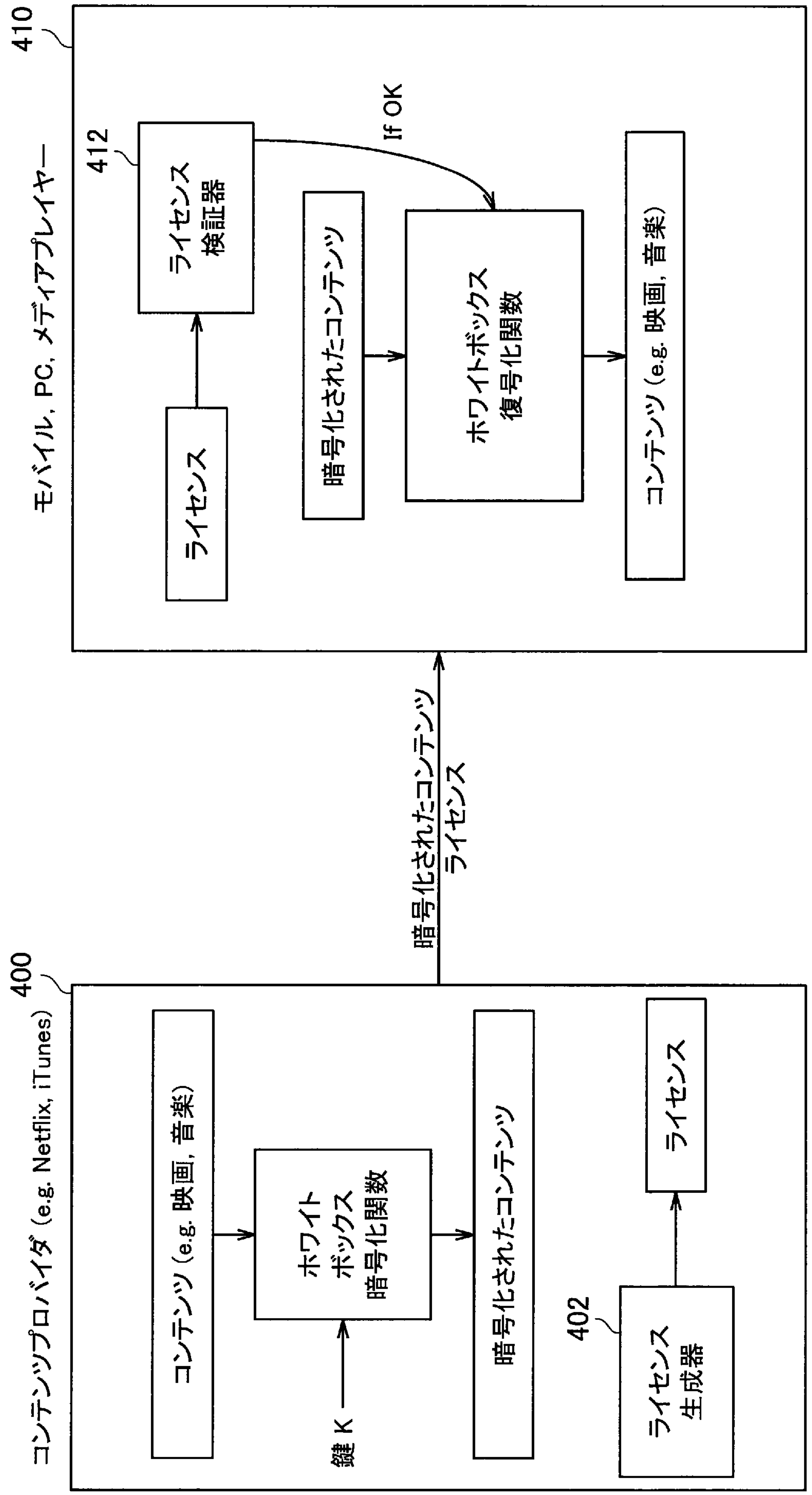


$$\begin{pmatrix} r_{A\,in}^{r} \\ r_{B\,in}^{r} \\ r_{C\,in}^{r} \\ r_{D\,in}^{r} \end{pmatrix} = M \cdot \begin{pmatrix} r_{A\,out}^{r-1} \\ r_{B\,out}^{r-1} \\ r_{C\,out}^{r-1} \\ r_{D\,out}^{r-1} \end{pmatrix}$$

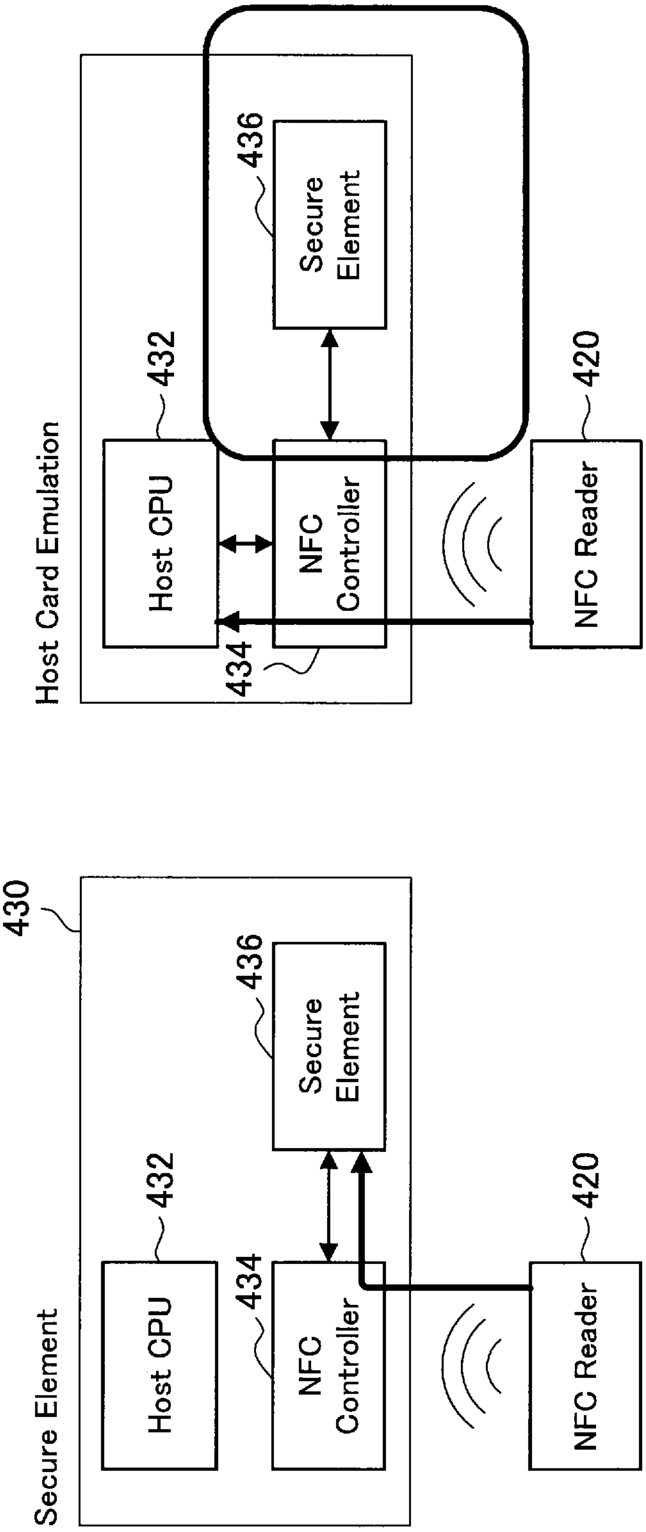
［図31］


410
ユーザデバイス (e.g. PC)
Media player application
C
復号
P
鍵 K
whitebox技術
コンテンツ配信 C
400
コンテンツサーバ (クラウド)
P: プレミアムコンテンツ (映画)
P
暗号化
C
鍵 K

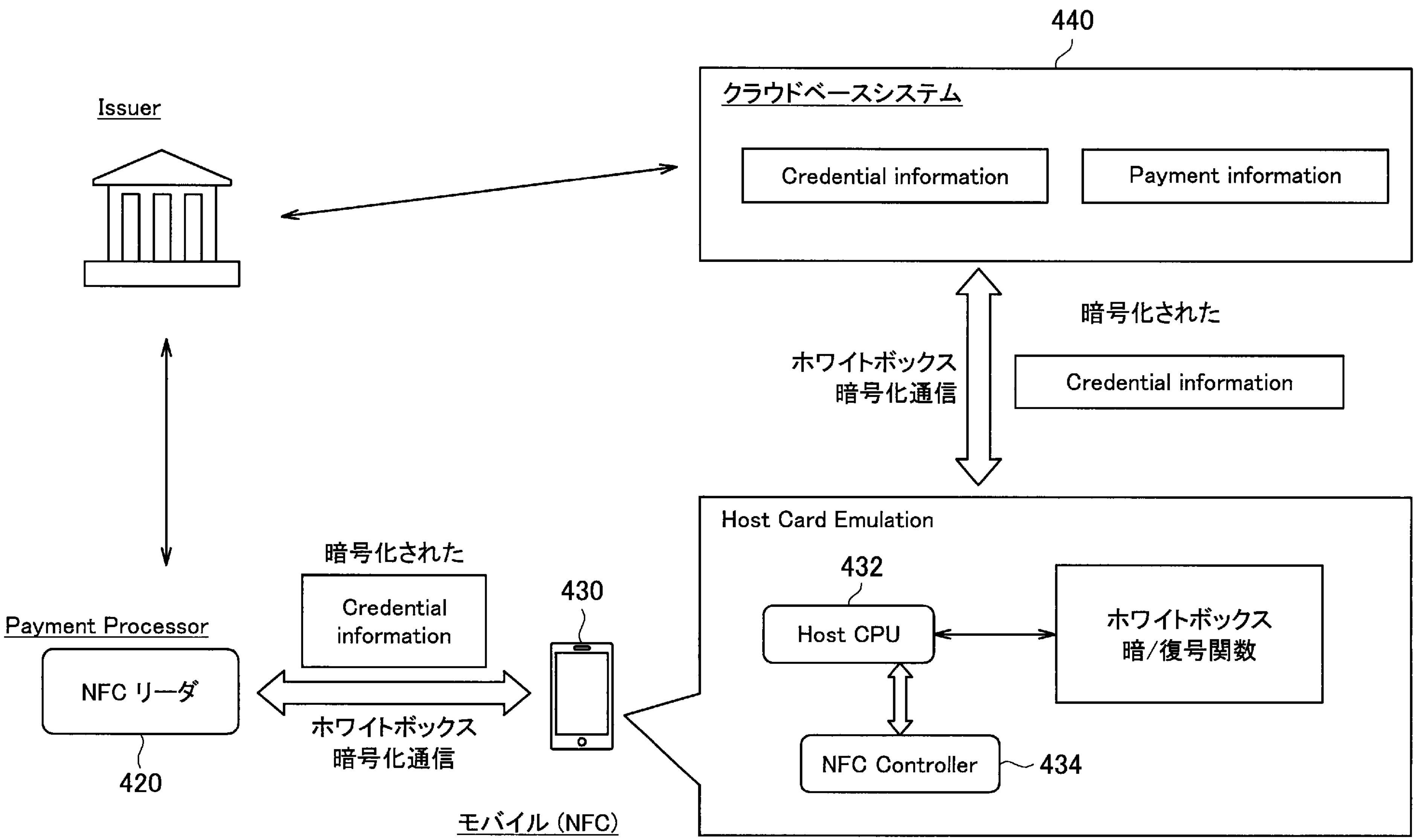
［図32］


410
モバイル，PC，メディアプレイヤー
412
ライセンス検証器
If OK
ライセンス
暗号化されたコンテンツ
ホワイトボックス復号化関数
コンテンツ（e.g. 映画，音楽）
暗号化されたコンテンツ
ライセンス
400
コンテンツプロバイダ（e.g. Netflix, iTunes）
コンテンツ（e.g. 映画，音楽）
ホワイトボックス暗号化関数
鍵 K
暗号化されたコンテンツ
ライセンス
402
ライセンス生成器

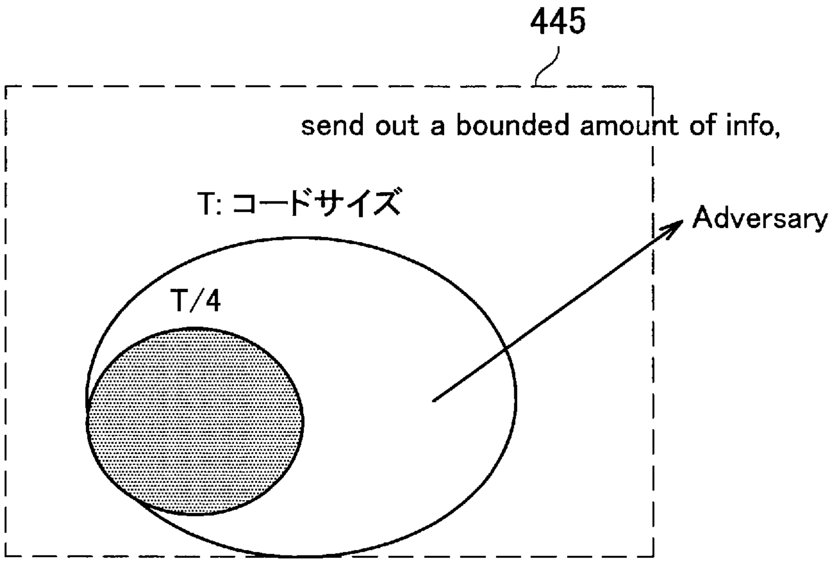
[図33]


Host Card Emulation
Host CPU
432
NFC Controller
434
Secure Element
436
NFC Reader
420
Secure Element
430
Host CPU
432
NFC Controller
434
Secure Element
436
NFC Reader
420

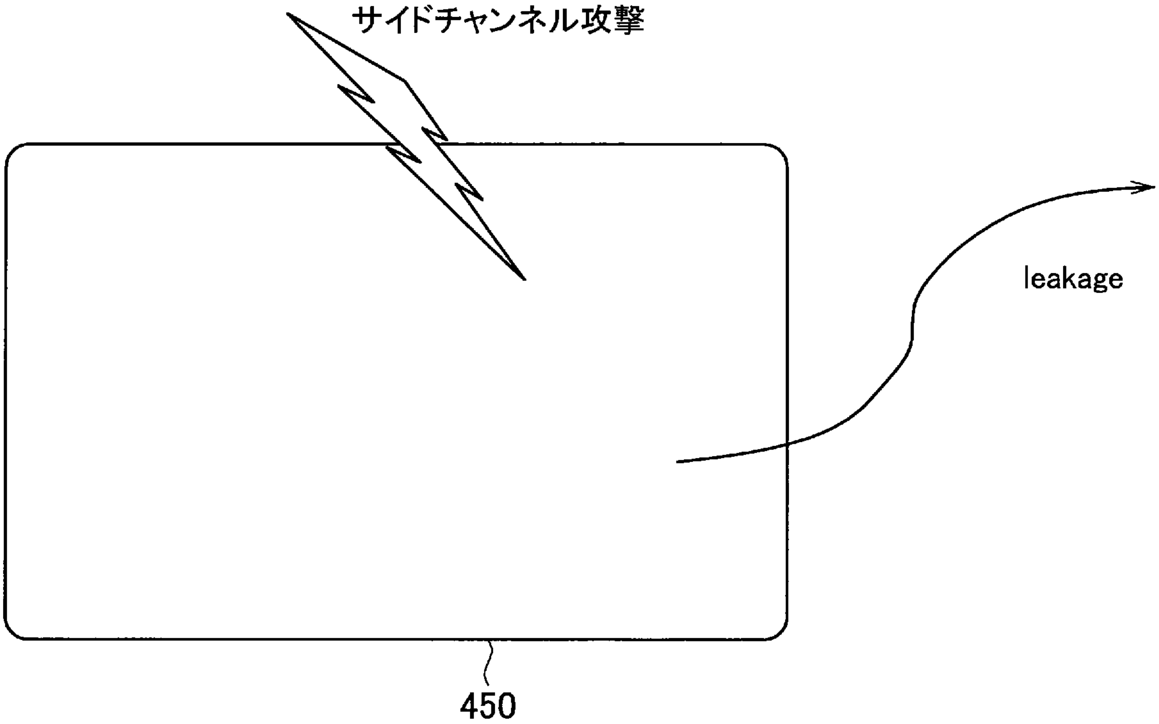
[図34]


440
クラウドベースシステム
Credential information
Payment information
Issuer
暗号化された
Credential information
ホワイトボックス
暗号化通信
Host Card Emulation
432
Host CPU
ホワイトボックス
暗/復号関数
NFC Controller
434
Payment Processor
NFC リーダ
420
暗号化された
Credential
information
ホワイトボックス
暗号化通信
430
モバイル (NFC)

[図35]


445
send out a bounded amount of info,
T: コードサイズ
T/4
Adversary

[図36]


サイドチャンネル攻撃
leakage
450

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2018/020341

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl. G09C1/00(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
Int.Cl. G09C1/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Published examined utility model applications of Japan	1922-1996
Published unexamined utility model applications of Japan	1971-2018
Registered utility model specifications of Japan	1996-2018
Published registered utility model applications of Japan	1994-2018

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2017/0141911 A1 (NXP B. V.) 18 May 2017, paragraphs [0068]-[0080] & EP 3169017 A1 & CN 106953723 A	1-18
A	BOGDANOV, A., ISOBE, T., White-box cryptography revisited: space-hard ciphers, Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, ACM, 12 October 2015, pp. 1058-1069	1-18

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search
07.08.2018

Date of mailing of the international search report
21.08.2018

Name and mailing address of the ISA/
Japan Patent Office
3-4-3, Kasumigaseki, Chiyoda-ku,
Tokyo 100-8915, Japan

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2018/020341

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2012-520589 A (IRDETO CORPORATE B.V.) 06 September 2012, paragraphs [0127]-[0261] & US 2012/0002807 A1, paragraphs [0110]-[0134] & WO 2010/102960 A1 & EP 2406916 A1 & CA 2754094 A & CN 102461058 A & KR 10-2012-0030335 A	1-18
P, A	JP 2017-187724 A (SONY CORPORATION) 12 October 2017, paragraphs [0001]-[0074] & US 2017/0294148 A1, paragraphs [0001]-[0118]	1-18

国際調査報告　　国際出願番号　ＰＣＴ／ＪＰ２０１８／０２０３４１

Ａ．　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Int.Cl.　G09C1/00(2006.01)i

Ｂ．　調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

Int.Cl.　G09C1/00

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	１９２２－１９９６年
日本国公開実用新案公報	１９７１－２０１８年
日本国実用新案登録公報	１９９６－２０１８年
日本国登録実用新案公報	１９９４－２０１８年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

Ｃ．　関連すると認められる文献

引用文献の カテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
A	US 2017/0141911 A1 (NXP B. V.) 2017.05.18, 段落 [0068] － [0080] & EP 3169017 A1 & CN 106953723 A	1-18
A	BOGDANOV, A. and ISOBE, T., White-box Cryptography Revisited: Space-Hard Ciphers, Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, ACM, 2015.10.12, Pages 1058-1069	1-18

☑　Ｃ欄の続きにも文献が列挙されている。　　□　パテントファミリーに関する別紙を参照。

＊　引用文献のカテゴリー

「Ａ」特に関連のある文献ではなく、一般的技術水準を示すもの

「Ｅ」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「Ｌ」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）

「Ｏ」口頭による開示、使用、展示等に言及する文献

「Ｐ」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献

「Ｔ」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「Ｘ」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Ｙ」特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「＆」同一パテントファミリー文献

国際調査を完了した日
０７．０８．２０１８

国際調査報告の発送日
２１．０８．２０１８

国際調査機関の名称及びあて先
日本国特許庁（ＩＳＡ／ＪＰ）
郵便番号１００－８９１５
東京都千代田区霞が関三丁目４番３号

特許庁審査官（権限のある職員）　５Ｓ　６３０４
行田　悦資
電話番号　０３－３５８１－１１０１　内線　３５４６

Ｃ（続き）．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求項の番号
A	JP 2012-520589 A（イルデト・コーポレート・ビー・ヴイ） 2012.09.06, 段落［0127］－［0261］ & US 2012/0002807 A1, 段落［0110］－［0134］ & WO 2010/102960 A1 & EP 2406916 A1 & CA 2754094 A & CN 102461058 A & KR 10-2012-0030335 A	1-18
P, A	JP 2017-187724 A（ソニー株式会社） 2017.10.12, 段落［0001］－［0074］ & US 2017/0294148 A1, 段落［0001］－［0118］	1-18